AUTOMATED FAULT TREE ANALYSIS OF A FAST BREEDER PRIMARY LOOP

by

MANOJ KUMAR

NUCLEAR ENGINEERING AND TECHNOLOGY PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1987

AUTOMATED FAULT TREE ANALYSIS OF A FAST BREEDER PRIMARY LOOP

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

MANOJ KUMAR

to the

NUCLEAR ENGINEERING AND TECHNOLOGY PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1987

CERTIFICATE

This is to certify that the work presented in this thesis entitled, "AUTOMATED FAULT TREE ANALYSIS OF FAST BREEDER PRIMARY LOOP", has been carried out under our supervision and has not been submitted elsewhere for the award of degree.

K. Sri Ram
(K. Sri Ram)
Professor

Prabhat Munshi
(P. Munshi)
Lecturer

Nuclear Engineering & Technology
Indian Institute of Technology
Kanpur-208016

December, 1986.

ACKNOWLEDGEMENTS

I am highly grateful to my guides Prof.K.Sri Ram and Mr. P. Munshi for their timely help and encouragement. But for their active participation and cooperation, this thesis work would not have been a success. It was only their affection and understanding that instilled a keen interest in my inner self and as a result I could work to the best of my abilities given the other constraints. They command high respect and regard in my heart.

My thanks are also due to my other teachers Prof. Sengupta and Dr. Kalra for maintaining high academic standards and helping me and my class mates as and when necessary.

I have high praise for other department staffs - Mr. Pathak, Mr. Tomar, Mr. Yadav and Mr. Gopal. They were highly cooperative and helpful.

I can not but praise my class-mates for their selfless attitude and active cooperation. It was only due to them that we could enjoy throughout the M.Tech. Program.

- MANOJ KUMAR

CONTENTS

ABSTRACT

We ventured to develope a complete software package for fault tree analysis of any system be it mechanical, electrical or nuclear. In doing so attempt was made to improve upon the existing computer codes in terms of ease to use, CPU time, efficiency and software facilities. A particular attention was paid to enlarge the scope of the software. For example, almost all kinds of gates have been allowed using fault tree modification program. In fault tree quantification, 'M' out of 'N' type of system has been allowed and a new formulation has been done using Fussell's approximation. Save fault tree construction, all other processes in the fault tree analysis has been automated including the electrical systems associated in nuclear systems. We got unavailability at the end of 40 years to be $0.5x10^{-7}$ for SNR-300 primary loop which is satisfactory according to 10 CFR 100 criterions. We also established that increase of redundancies improves the system but some components being more critical than others. Our computer codes can, hence, be of immense help in system design.

CHAPTER 1

INTRODUCTION

The present work is devoted to generate a complete software package for reliability analysis of any system using fault tree method in a most general form. Also discussed is the application of the package to primary coolant loop with emergency core cooling of a LMFBR for which model chosen is that of SNR-300 (A German 300 MW Fast Breader Reactor). This package can be compared favourably with earlier similar packages in terms of its facilities, methodology, efficiency and CPU time on a computer.

The first program finds minimal cutsets for a fault tree containing upto 100 OR gates or 100 basic events. This number could be increased easily by increasing dimension of the concerned variables. MOCUS[3], on the other hand, can be used to find minimal cuts for upto 20 gates in a given tree. In this new program PCOMCP all basic events are coded by prime numbers and all logical operations and minimization processes are carried out by simple arithmetic operations. This reduces the storage requirement greatly since no character reading or logical operations are needed. A new method of providing input has been used in which number of lines or cards is no more than the number of basic events; thus simplifying the input process a great deal. In most of other programs input is given gate by gate.Thus, number of input cards equals number of gates and hence a lengthy process. As in MOCUS, the maximum order of cutsets can also be specified in the input data.

Second program quantifies the fault tree, i.e. finds top event unavailability (probability) and other related parameters. This program is based on KITT [3] formulation but there is an improvement. The present program takes care of M out of N systems for basic events and top event. It is much easier to specify the values of M & N rather than modifying fault tree itself to take into account such systems. This kind of systems are mostly encountered due to large redundancy in nuclear systems to improve reliability. As in KITT1 and KITT2 bracketing procedure has been used to find maximum and minimum top event unavailability. In present program, at some places FUSSEL's approximation has been used when such parameter are not accurately desirable and formulation for exact value is otherwise difficult. Data could be provided both in terms of Lambda and Meu or Q (unavailability) and w (unconditional failure intensity). In latter case by bisection method of iteration,Lambda and Meu are obtained. There are three options for providing accuracy desired. Option 0 ⇒ take all cutsets into account. Option 1 ⇒take maximum order of cutsets (specified).Option > 1 ⇒ specify the % accuracy desired. Integration uses Simpson's rule in which decional place accuracy is specified. It is obvious that program is most versatile and flexible in use.

Failure data and typical repair data for basic components have been obtained from WASH - 1400 report[1,2] Nuclear Engg. & Design (1984 V 81-82, NED 1984 V 83), Reliability technology by Green and Bourne[9] and other sources. From the analysis an unavailability of $0.5x10^{-7}$ has been obtained at the end of 40 year and it is seen that with time it decreases but not at a large rate for the primary coolant loop of LMFBR. This low unavailability has been obtained in LMFBR by using sufficient redundancies and standby systems.

Third program finds minimal cutsets for an electrical systems which contain large number of feedbacks, interconnection, and are represented by a ckt. graph rather than a fault tree. Entirely different approach is needed in this case. Electrical systems are integral part of nuclear power plant hence the interest. The program finds basic minimal path and the combinations which break these paths constitute the minimal cutsets [Chapter 6] .

Last program is a fault tree modification program. It modifies the fault tree to take care of XOR, NOT, NAND, NOR gates. It utilises the gate equivalence for XOR and De Morgan's law for others. In some cases, level of the tree increases which makes the program slightly complex.

This kind of facility is not available in MOCUS or PREP-KITT. 'M' out of 'N' kinds of gates are taken care of at quantification level. Output of this program is a fault tree containing only AND and OR gates. Some of the basic events occur in negated form. It first determines the sub-branches which necessitates increase in the level of tree. All the basic event entries are modifies accordingly. GATE equivalence and De Morgan's law suggest the rest of the procedure.

CHAPTER 2

SYSTEM DESCRIPTION AND MODELLING

The system we chose in this study for reliability analysis was primary sodium loop and emergency core cooling system of SNR-300 which is also called Kalkar nuclear power station (a joint German, Belgium and Dutch project). This design is very similar for many fast breeders and as we will see the main features are also valid for Clinch-River Breeder reactor plant and other fast breeder reactors.

2.1 SNR 300 Details (Only Primary & Intermediate Loop)[7]

The three parallel loops of the heat transfer system (HTS) transfer the heat produced in the reactor via intermediate heat exchangers and steam generators to the electricity producing system. They are furthermore used for the decay heat removal with the pump running at 5% of their nominal speed. Even with two loops out of service, natural convection in the third loop can remove the decay heat without reaching boiling at the core outlet. Even in cases leading to a failure of all main loops, the decay heat removal can take place through six emergency heat exchangers operating in parallel and located inside the reactor tank.

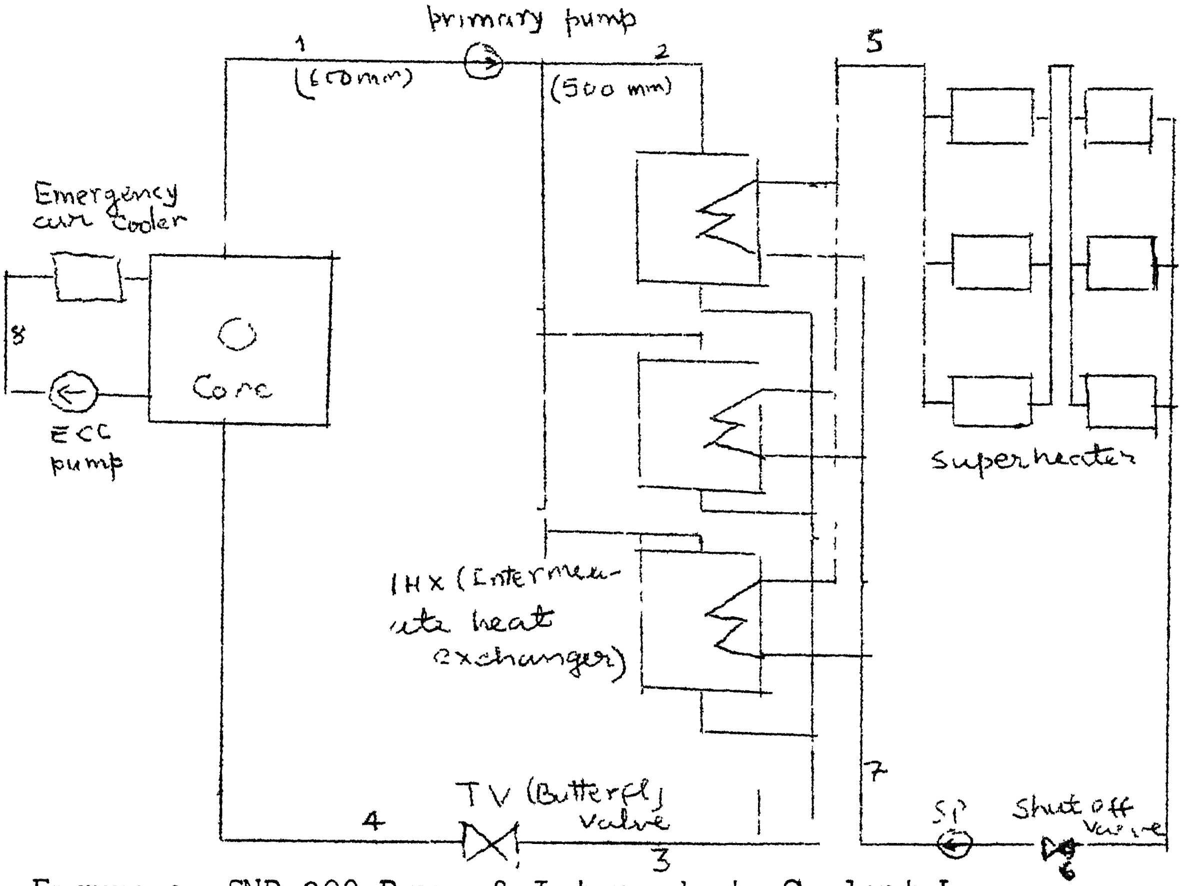


Figure : SNR-300 Prim. & Intermediate Coolant Loop

The pipes and components of heat transfer systems with the exception of the steam generator are using unstabilized austenitic stainless steel X8 Cr Ni 1811 (WN1.4948).

Each primary loop contains one circulating pump, a bank of three intermediate heat exchangers, one sodium flow meter and one throttle valve to reduce the coolant flow following a reactor shut down.

The coolant circulating pumps are positioned in the hot leg to provide a sufficient suction head at a low excess pressure prevailing in the upper part of the reactor tank. The pumps are of variable speed, single stage, single-flow, radial type, arranged vertically with free sodium surface and argon cover gas atmosphere.

Heat exchangers are of straight tube design with a floating head at the lower end with the primary sodium flowing on the shell side, an arrangement that allows the tube bundle to be withdrawn without cutting the primary loop pipes.

The pipe connecting the reactor vessel to the pump has a diameter 600 mm which reduces to 500 mm for all other main pipes.

A butterfly valve located between IHX and the reactor vessel limits the sodium flow rate during decay heat removal operation, thus reducing thermal shocks.

All pipes except those in the annular region surrounding the reactor vessel are provided with an electric trace-heating system. The pipes in the reactor vessel, the pumps and IHX's are preheated by hot nitrogen.

The pipes of the three primary loops are laid out symmetrically in the reactor cell, from which they run to the three parallel primary cells. This arrangement eases shielding of the piping penetrations and reduces the activation of the structural material and of the secondary sodium.

2.2 Fault Tree Construction

First of all referring to SNR-300 figure we divide loop into two parts.

(a) Primary Loop - Consisting of pipe segments 1,2,3,4, primary pump, Butterfly valve and heat exchanger.

(b) Secondary Loop - Consisting of pipe segments 5,6,7, shut off valve, superheater, evaporator and secondary pump.

Then following observations were made·

(i) Pump has two kinds of failures - (a) single ended or minor failure; (b) double ended or major failure.

(ii) Different pipe segments can in general (and in fact they are so) be of different diameters hence their failure data is expected to be different from one another. As a result, these pipe segments break or leak must be considered as separate basic events.

(iii) Heat exchangers, Butterfly valves, primary pump, secondary pump, shut off valve, superheater, evaporator have redundant duplications intended to improve reliability.

(iv) Only when primary or intermediate heat transport system and emergency core cooling system fail, there is uncontrolled rise in core temperature.

(v) Since major components have redundancies, there must be maintenance when one or more of redundant and main component fail. Hence repair rate must be taken.

(vi) Primary pump is a sodium pump and is more sophisticated. Its data must be chosen carefully.

(vii) There are three identical primary loops hence over all system is 1 out of 3.

Based on above information we make further subdivision of the system as follows: Primary loop is divided into two parts, (a) Valve line consisting of pipe segments 3,4 and Butterfly valve, and (b) Pump line consisting of pipe segments 1,2, primary pump and intermediate heat exchangers. Any of these parts' failure will lead to primary loop failure. Valve line failure occurs when either there is no supply from Butterfly valve or pipe length 4 has failed. Butterfly valve supply is cut off when either pipe length 3 is cut off or Butterfly valve is stuck. Pump line failure occurs when either there is no input to heat exchanger or heat exchanger fails. There is no input to heat exchanger if either pipe segment 2 fails or no supply from primary pump. There is no supply from primary pump if pipe segment 1 fails or pump fails.

Intermediate loop fails if either valve line (consisting of pipe length 6 and 7, shut off valve and sec. pump) fails or super heater, evaporator line fails (consisting of superheater, evaporator or pipe length 5). Valve line fails if either pipe length 6 fails or sub-system from valve to heat exchanger fails. The later fails if pipe length 7 fails or pump and valve system fails. Pump and valve system fails if either of these fails. Superheater, evaporator line fails if pipe length 5 fails or any of superheater and evaporator fails.

Emergency core cooling system fails if either air cooler fails or there is no supply to air cooler. There is no supply to air cooler if either emergency pump fails or pipe length fails.

2.3 Data to be Used

Based on Appendix I, we decided to use the following data:

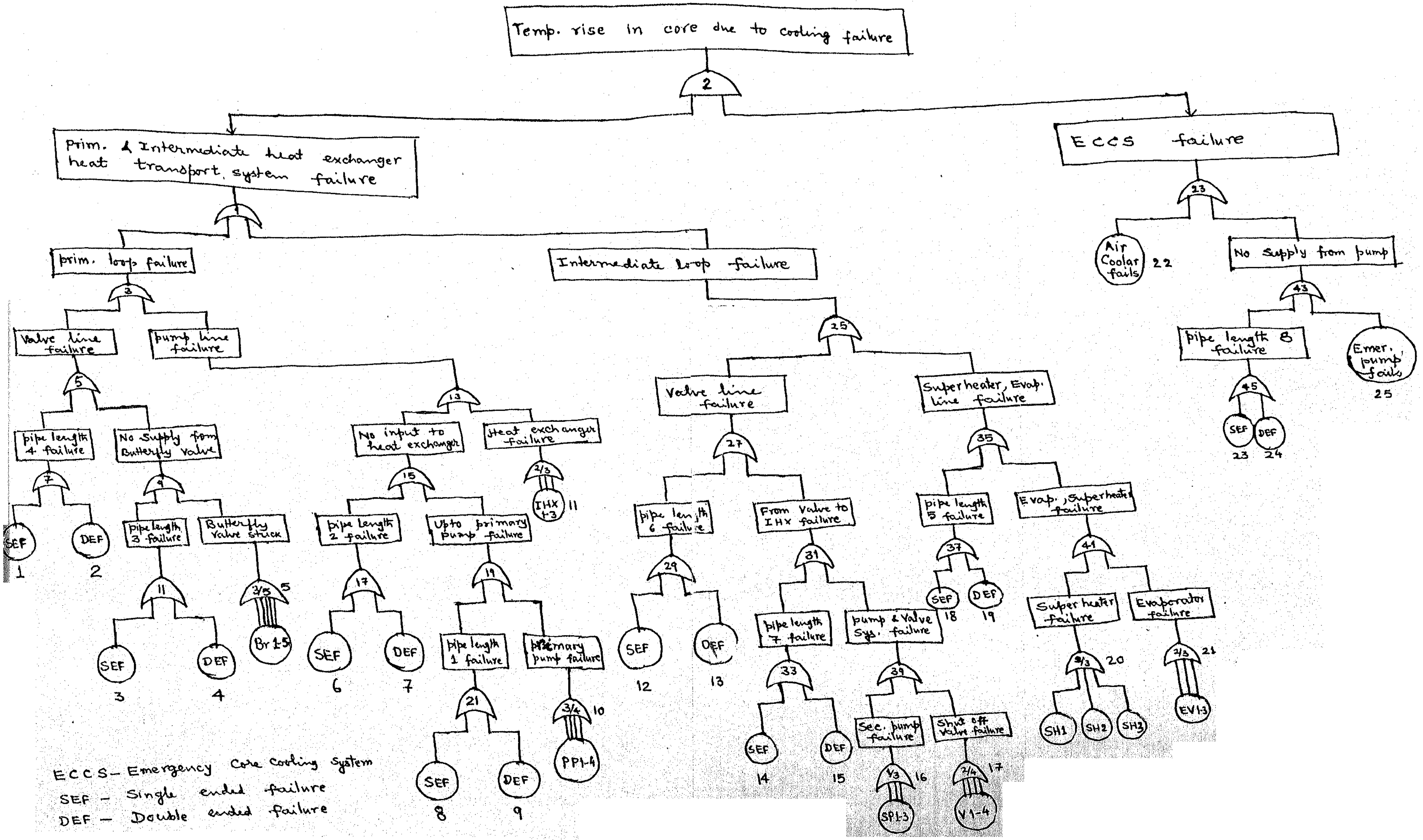
Component	λ hr^{-1}	μ hr^{-1}
Emer. cooling pipe (single ended failure)	$2.0x10^{-9}$	$1.2x10^{-7}$
Emer.cooling pipe (double ended failure)	$2.0x10^{-10}$	$1.0x10^{-8}$
Other pipes (single ended)	$1.0x10^{-10}$	$5.0x10^{-9}$
Other pipes (double ended)	$1.0x10^{-12}$	$4,0x10^{-11}$
Valve (Butterfly)	$1.6x10^{-5}$	$1.0x10^{-4}$
Primary pump	$9.17x10^{-7}$	$1.0x10^{-5}$
Heat exchanger	$4.184x10^{-6}$	$2.0x10^{-5}$

Component	λ hr^{-1}	μ hr^{-1}
Secondary pump	2.5×10^{-6}	2.5×10^{-5}
Shut off valve	3.0×10^{-5}	3.0×10^{-4}
Super-heater	8.0×10^{-6}	5.0×10^{-5}
Evaporator	8.0×10^{-6}	5.0×10^{-5}
ECC pump	2.5×10^{-6}	2.5×10^{-4}
Air cooler	3.0×10^{-6}	5.0×10^{-4}

2.4 Reliability Program Objectives

We aim at ensuring that the liklihood of exceeding the nuclear radiation dose guidelines of 10CFR 100[12] at the plant site boundary should not be greater than 10^{-6} per reactor operating year that in-place core-coolable geometry will be lost [8]. Loss of in-place coolable core geometry is a failure criterion which is used to characterize very low probability events. If core coolable geometry is lost, there is no assurance that significant core damage could not occur, even though there is still a low probability that site boundary dose guidelines could be exceeded since the plant includes a number of containing barriers, one of which is the containment structure. The probability of occurrence of potential initiators of loss of coolable geometry can be controlled by design. It then becomes the task of the Reliability Program to assure the high reliability of systems necessary to prevent the onset of such initiating events. Basic tasksof the Reliability Program are (1) to identify those extremely unlikely events having the potential to produce loss of coolable core geometry (2) to ensure through reliability engineering design that all such events are of sufficiently low probability to meet the goal set (3) to confirm this high reliability through analytic assessment and testing.

An initial reliability allocation against the overall goal was made for potential initiators which might threaten coolable geometry [8]. The division of the overall numeric goal among safety systems was based on the functional role of each system and its predicted relative failure potential. On the basis of this, the preliminary unreliability allocation established for shutdown heat removal system [8] was $< 8\text{x}10^{-7}$ failures per year.



ECCS – Emergency Core cooling system

SEF – Single ended failure

DEF – Double ended failure

2.5 FAULT - TREE

CHAPTER 3

PROGRAM PCOMCP FOR FINDING MINIMAL CUTSETS

3.1 Definition

Cutsets – A cutset is a collection of basic events; if all these basic events occur, the top event is guaranteed to occur.

Pathset - It is a collection of basic events and if none of the events in the set occur, the top event is guaranteed to not occur.

Minimal cutset - A minimal cutset is such that if any basic event is removed from the set, the remaining events collectively are no longer a cutset. This is obtained from cutset by removing cuts which are redundant according to some rules laid down below.

Minimal Pathset - is a path set such that if any basic event is removed from the set, the remaining events collectively are no longer a path set.

3.2 Redundancies to be removed for Generation of Minimal Cutset

(i) Redundant Factor

If in any cut we have A.A, A or A like term replace it by A.

(ii) Subset Redundancy

If two cuts are such that one is subset of other remove the superset, i.e. A + AB = A.

(iii) Term Redundancy

If two terms are repeated in the cutset then remove the repeated term, AB + AB = AB.

3.3 Algorithm

The program is based on the unique factorization properties of natural numbers stated in the form of following theorem.

Unique Factorization Theorem [4]

Every natural number greater than 1 can be expressed as a product of prime factors in one and only one way, apart from the order in which the factors are written.

Our strategy is to assign one prime number to each basic event, doing arithmetic operations for logical GATES, removing redundancies in cutsets and finally decode them back.

A particular combination of basic events can be expressed uniquely as a single number which is equal to the product of prime numbers corresponding to the basic events.

3.4 Program Details

We follow the following steps:

(i) Assignment of one prime number to each basic event.

(ii) Go bottom up.

(iii) Go up level by level (to be described). A level is the number of gates from top after which the event is encountered.

(iv) For each of OR gate encountered we preserve all the inputs to gate in an array.

(v) For each AND gate we multiply the input codes.

(vi) Of the final array, if two number is multiple of other larger or equal is removed. This is to remove of redundancy No. (ii) and (iii).

(vii) Of the final set, each numbers is prime factorized. If any factor isrepeated in the factor set they are removed. The factor set when decoded gives minimal cutset. This step removes the redundancy (i).

(viii) If number of prime factors increase the maximum order of cutsets wanted, that cut is also removed.

(ix) AND gate is designated by an even number to the gate and OR gate by an odd number.

Method of inputting the tree structure is by assigning a series of integers which represent the gates linking the top event to a primary event. The length of each series is equal to maximum level (i.e. maximum number of gates from Top event to basic event) plus one. The gate number 0 indicates that there is no gates at that position. The order of row corresponds to the configuration of a fault tree.

Level by level is done as follows:

First line of input data is read and level is decided. The number is stored in the matrix element N(L,1) where L is level number. Second line of input data is then read. If level number is same as previous number, the last gate operation is done. If last gate is OR gate (i.e. gate number is odd), it is stored in N(L,2) i.e. in the N(L,x) which is vacant (i.e. zero) and x is minimum. On the other hand if gate is AND gate (i.e. gate number is even) all N(L,x) elements are multiplied by the current line basic event number. Once the gate operation is done level is reduced by one and this number is stored. Now the third line is read. If the level number equals the level number stored, the gate operation is done according to rules aforesaid and gate type indicated by the gate number being odd or even at the level's place from the Top gate number row wise. Level number is reduced by one and this level is stored. In cases when the current level is not equal to the stored level number, the current basic event coded number is stored in N(L',1) and next line read, if level is again not equal to previous level number, this basic even number is stored in N(L",1) and this is carried till the two successive levels are equal. Gate operation is done and level number is again reduced. If this level pointer is equal to previous level number, proper operation done and this process is repeated till all the numbers are in N(1,x) where x varies from 1 to No. of OR gates in the fault tree.

To remove the redundancies (ii) and (iii), the final set of numbers N(1,x) $\forall$ x <No. of OR gates, is checked for multiplicity. If any number in this set is multiple of any other, the larger (or equal) is set equal to zero. Once this procedure is carried out supersets are removed. Next for each non-zero elements in N(1,x) set, prime factors are found. If any prime factor is repeated in any element in N(1,x), this repeated factor is removed, i.e. one of repeated factors are kept. This process ensures the removal of type (i) redundancy. Corresponding to each number in set N(1,x), the set of non-repeated prime factors when decoded give minimal cutset elements. The number of minimal cutsets equal the number of nonzero element in set N(1,x).

3.5 Salient Features of the Program

(i) Coding by prime numbers ease out the minimization of cutset procedure.

(ii) Logical operations carried out by arithmatic and matrix operation.

(iii) Actual basic event No. when coded by prime numbers does not increase the CBE value very much as compared to actual basic event No.(CBE=Coded Basic Event).

(iv) The maximum order of cutsets could be specified.

(v) Level by level operation is actual practice reduces the total number of operation required.

(vi) Because of (i), (iii) and (v) the program is efficient and takes less CPU time.

(vii) Memory storage required is less than many other minimal cutset enumeration programs.

(viii) The maximum number of OR gates and input right now could be 100. This can be increased by simply increasing the dimension of some arrays in the program.

3.6 Limitations of the Program

(i) All gates must have only two inputs. Only the top gate can have 3 or more inputs.

(ii) As itis the program takes care of only AND and OR gates. Some more gates can be taken into account as mentioned below:

(a) NOT gate - the basic event itself is changed. It is considered independent event with R = 1 - not inverted event reliability.

(b) INHIBIT Gate - It is considered an AND gate. With conditional event as a basic event.

(c) Priority AND - It is basically AND gate with some sequencing; so it is treated as AND gate.

3.7 Operations to be Carried Out

The operation can be summarized as below:

(a) A line is read and level is decided.

(b) If level is equal to previous step level no. Logical operation is to be done otherwise CBE is stored in N(L,1).

(c) Logical operation is carried out as follows: If it is an OR gate the CBE is to be stored in vacant place of N(L,x). On the other hand, if it is an AND gate all the elements of N(L, x) are multiplied by CBE,

(d) If any logical operation is done LP is decreased by 1. If NG (LP) is even,(i.e. AND gate) all combination of multiplication of N(LP, x) and N(LP + 1, x) is carried out and stored in N(LP, x). In cases when NG (LP) is odd, All elements of N(LP + 1,x) stored in vacant (i.e. zero) places of N(LP, x). All elements of N(LP + 1, x) reduced to zero.

(e) LP decreased further and (d) line operation carried out till LP is zero.

(f) When all of input data is over, factorization, minimization and decoding is done as outlined above. If data is not over go to (a).

CHAPTER 4

SYSTEM QUANTIFICATION - A THEORETICAL OVERVIEW

4.1 Some Definitions [13] and Symbol Meanings

(a) Reliability at time t = R(t).

The probability that the component experiences no failure during the time interval (O,t] given that the component was repaired at time zero.

(b) Unreliability at time t = F(t) = 1-R(t) (4.1)

(c) Failure density at time t = f(t)

The first order derivative of F(t).

(d) Failure rate = r(t) .

The probability that the component experiences a failure per unit time at time t given that the component was repaired at time zero and has survived to time t.

(e) Mean time to failure = MTTF

The expected value of time to failure, i.e. mean of the span of time from repair to first failure

$$\text{MTTF} = \int_{0}^{\infty} t\, f(t)\, dt \qquad (4.2)$$

(f) Repair probability at time t = G(t).

The probability that the repair is completed before time t given that the component failed at time zero.

(g) Repair density of G(t) = g(t)

The first order derivative of G(t).

(h) Mean time to repair = MTTR.

The expected value of time to repair, i.e. mean of length of time from the failure to the succeeding first repair

$$\text{MTTR} = \int_0^\infty t\, g(t)\, dt \qquad (4.3)$$

(i) Repair rate = m(t).

The probability that the component is repaired per unit time at time t given that the component failed at time zero and has been failed to time t.

(j) Availability at time t = A(t).

The probability that the component is normal at time t given that it was as good as new at time zero.

$$A(t) \geq R(t)\ .$$

(k) Unavailability at time t = Q(t)

The probability that a component is in the failed state at time t, given that it jumped into the normal state at time zero

$$Q(t) \leq F(t)$$

(l) Conditional failure intensity = $\lambda(t)$

The probability that the component fails per unit time at time t, given tnat it is in the normal state at time zero and is normal at time t.

(m) Unconditional failure intensity at time t = w(t)

The probability that a component fails per unit time at t, given that it jumped into the normal state at time zero.

(n) Expected number of failures (ENF) = W(t, t+dt)

Expected number of failures during [t, t+dt), given that the component jumped into the normal state at time zero.

(o) $W(t_1, t_2)$ = ENF over a period.

Expected number of failures during (t_1, t_2), given that the component jumped into the normal state at time zero.

$$W(t_1, t_2) = \int_{t_1}^{t_2} w(t)\, dt \tag{4.4}$$

(p) Conditional repair intensity = $\mu(t)$

The probability that a component is repaired per unit time at time t, given that it jumped into the normal state time zero and is failed at time t.

(q) Unconditional repair intensity at time t = v(t)

The probability that the component is repaired per unit time at time t, given that it jumped into the normal state at time zero.

(r) Expected number of repairs is an interval

= $V(t_1, t_2)$

Expected number of repairs during $[t_1, t_2)$, given that the component jumped into the normal state at time zero.

$$V(t_1, t_2) = \int_{t_1}^{t_2} v(t)\, dt \tag{4.5}$$

4.2 Fundamental Relations [14]

$A(t) = R(t)$ for non-repairable components

$\lambda(t) = r(t)$ for non-repairable components

$$R(t) = \int_t^\infty F(u)\, du \tag{4.6}$$

$$r(t) = \frac{f(t)}{1-F(t)} \tag{4.7}$$

$$R(t) = \exp\left[- \int_0^t r(u)\, du\right] \tag{4.8}$$

$$f(t) = r(t) \exp\left[- \int_0^t r(u)\, du\right] = r(t)R(t) \tag{4.9}$$

$$m(t) = \frac{g(t)}{1-G(t)} \tag{4.10}$$

$$G(t) = 1 - \exp\left[- \int_0^t m(u) du\right] \tag{4.11}$$

$$g(t) = m(t)[1-G(t)] = m(t) \exp\left[- \int_0^t m(u) du\right] \tag{4.12}$$

If constant failure rate $=\lambda= r(t)$ and non-repairable

$$\lambda(t) = - \frac{1}{R(t)} \frac{dR(t)}{dt} \tag{4.13}$$

$$R(t) = e^{-\lambda t} \tag{4.14}$$

$$F(t) = 1 - e^{-\lambda t} \tag{4.15}$$

$$f(t) = \lambda e^{-\lambda t} \tag{4.16}$$

$$MTTF = \frac{1}{\lambda} \tag{4.17}$$

If constant repair rate $=\mu= m(t)$

$$G(t) = 1 - e^{-\mu t} \tag{4.18}$$

$$g(t) = \mu e^{-\mu t} \tag{4.19}$$

$$MTTR = \frac{1}{\mu} . \tag{4.20}$$

4.3 Relations among the Whole Process

(a) Unconditional intensities w(t) and v(t). The component which fail during (t, t+dt] are of two types:

(i) Type 1: A Component which was repaired during [u, u+du), has been normal at time t, and fails during [t, t + dt), given that the component jumped into the normal state at time zero.

Probability for such component is v(t) du.f(t-u)dt since v(u)du = probability that the component is repaired during [u,u+du), given that it is as good as new at time zero and f(t-u)dt = the probability that the component has been normal to time t and failed during (t, t+dt], given that it was as good as new at time zero and was repaired at time u.

(ii) Type 2: A component which has been normal to time t and fails during [t, t+dt), given that it jumped into the normal state at time zero.

Probability of second type of components is f(t)dt w(t) dt = Probability that the component fails during [t,t+dt), given that it jumped into the normal state at time zero

$$w(t)dt = f(t)\, dt + dt \int_0^t f(t-u)\, v(u)\, du$$

or $$w(t) = f(t) + \int_0^t f(t-u)\, v(u)du \qquad (4.21)$$

(b) Second Relationship: The component which are repaired during [t, t+dt) is one which was failed during [u,u+du), has been failed to time t and repaired during [t, t+dt], given that the component jumped into the normal state at time zero.

The probability of such components is w(t) du.g(t -u)dt

$$v(t)\,dt = dt \int_0^t g(t-u)\,w(u)\,du$$

$$\therefore \quad v(t) = \int_0^t g(t-u)\,w(u)\,du \qquad (4.22)$$

(c) Unavailability:

$$Q(t) = W(0,t) - V(0,t) \qquad (4.23)$$

= Number of failures - Number of repairs at time t

(d) Failure intensity $\lambda(t)$.

$$\lambda(t) = \frac{w(t)}{1-Q(t)} \qquad (4.24)$$

(e) Repair intensity $\mu(t)$

$$\mu(t) = \frac{v(t)}{Q(t)} \qquad (4.25)$$

4.4 Laplace Transform Analysis for the Whole Process

Laplace Transform of (4.16)

$$L[f(t)] = \frac{\lambda}{S+\lambda}$$

$$L[g(t)] = \frac{\mu}{S+\mu} \quad (4.19)$$

Laplace transform of (4.21) and (4.22)

$$L[w(t)] = L[f(t)] + L[f(t)] . L[v(t)]$$

$$L[v(t)] = L[g(t)] . L[w(t)]$$

Using the L [f(t)] and L[g(t)]

$$L[w(t)] = \frac{\lambda}{S+\lambda} + \frac{\lambda}{S+\lambda}\, L\,[v(t)]$$

$$L[v(t)] = \frac{\mu}{S+\mu}\qquad L[w(t)]$$

Solving the two equations for L w(t) and L v(t)

$$L[w(t)] = \frac{\lambda\mu}{\lambda+\mu}\left(\frac{1}{S}\right) + \frac{\lambda^2}{\lambda+\mu}\left(\frac{1}{S+\lambda+\mu}\right)$$

$$L[v(t)] = \frac{\lambda\mu}{\lambda+\mu}\left(\frac{1}{S}\right) - \frac{\lambda\mu}{\lambda+\mu}\left(\frac{1}{S+\lambda+\mu}\right)$$

Taking Laplace inverse

$$w(t) = \frac{\lambda\mu}{\lambda+\mu} + \frac{\lambda^2}{\lambda+\mu}\; e^{-(\lambda+\mu)t} \qquad (4.26)$$

$$v(t) = \frac{\lambda\mu}{\lambda+\mu} - \frac{\lambda\mu}{\lambda+\mu}\; e^{-(\lambda+\mu)t} \qquad (4.27)$$

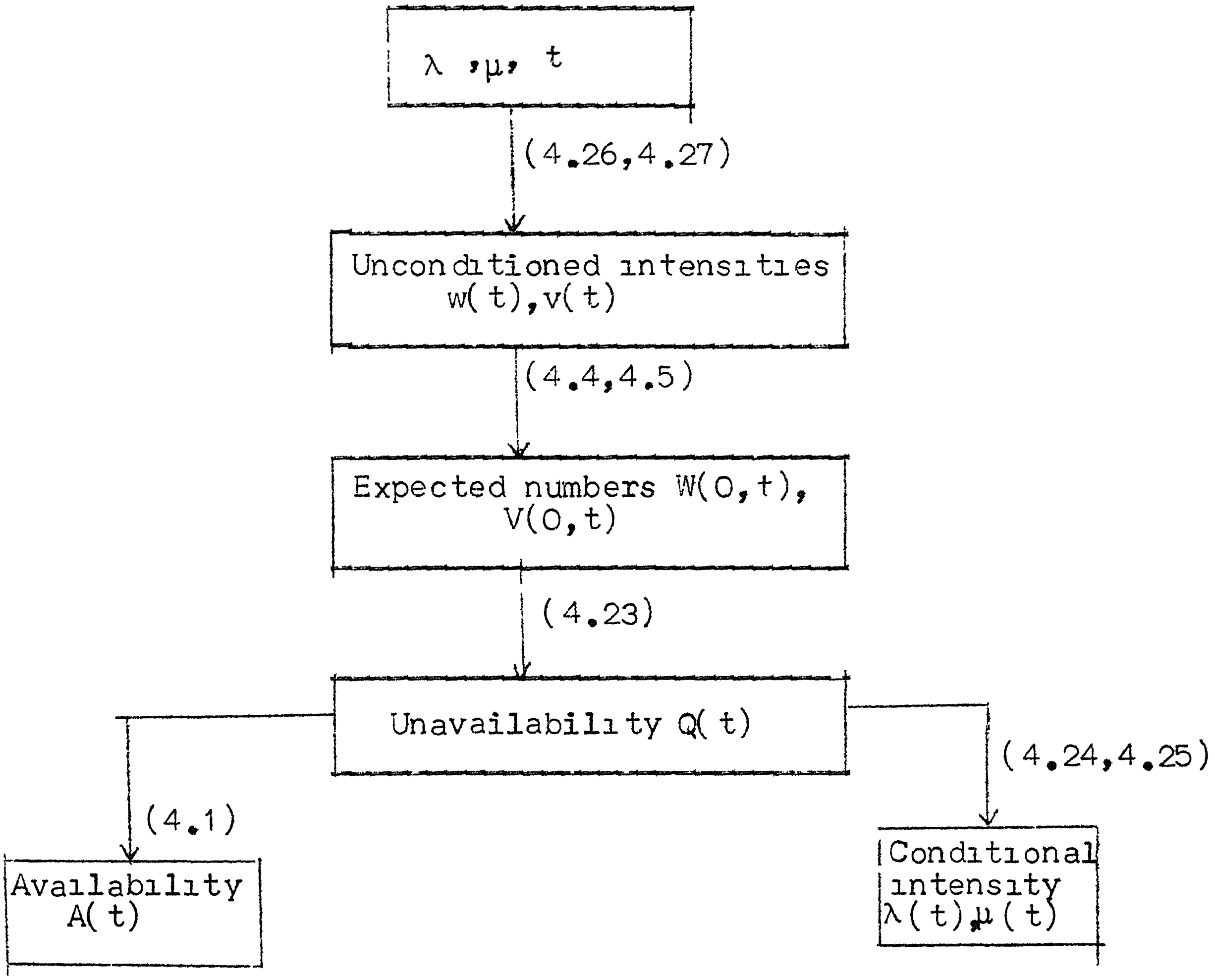
$$W(0,t) = \frac{\lambda\mu}{\lambda+\mu}\, t + \frac{\lambda^2}{(\lambda+\mu)^2}\,(1-e^{-(\lambda+\mu)t}) \qquad (4.28)$$

$$V(0,t) = \frac{\lambda\mu}{\lambda+\mu} - \frac{\lambda\mu}{(\lambda+\mu)^2}\,(1-e^{-(\lambda+\mu)t}) \qquad (4.29)$$

$$Q(t) = W(0,t) - V(0,t) = \frac{\lambda}{\lambda+\mu}\,(1-e^{-(\lambda+\mu)t}) \qquad (4.30)$$

4.5 System Analysis

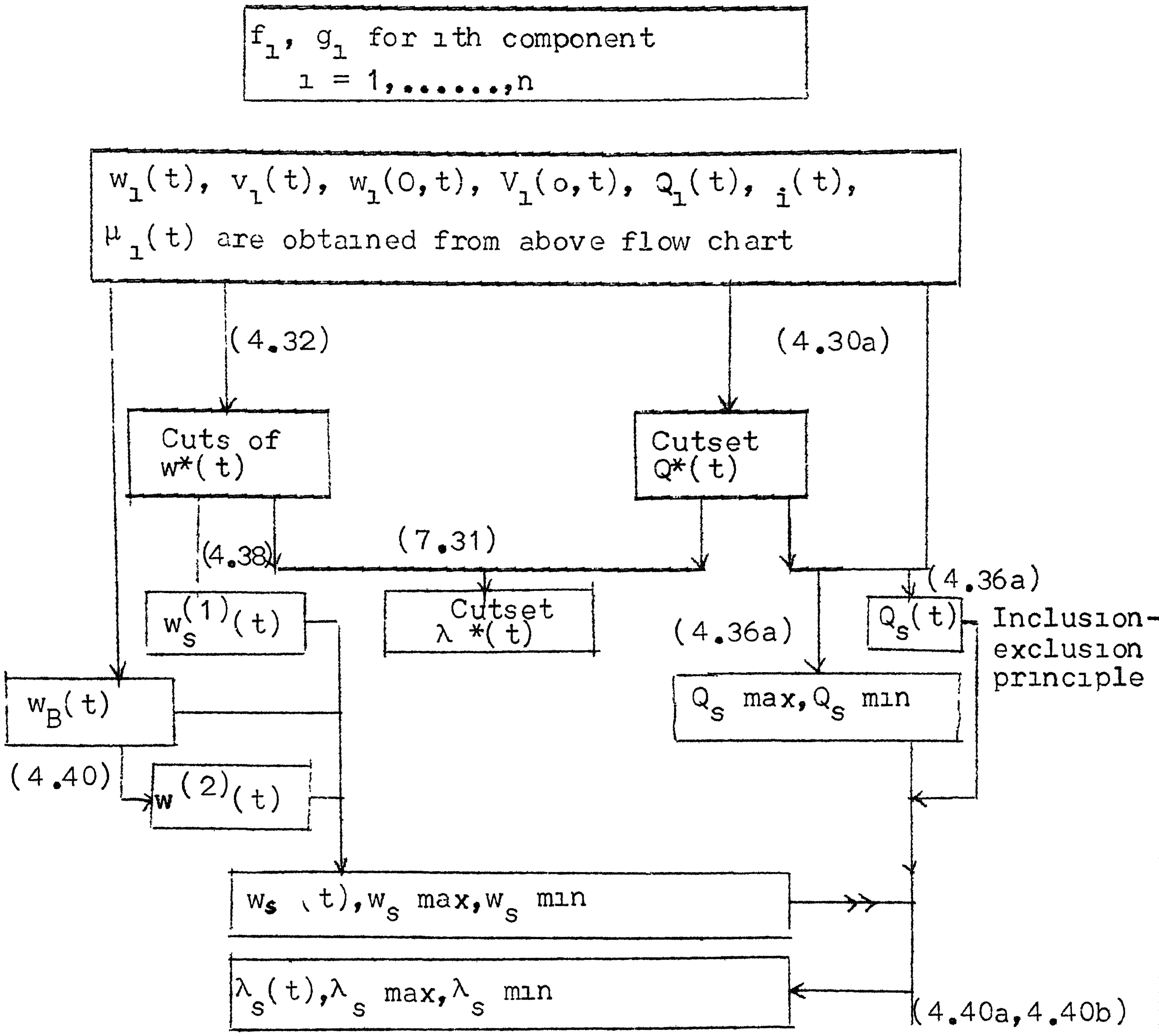
A. Component analysis is done on the basis of following flow chart:


λ ,μ, t
(4.26,4.27)
Unconditioned intensities w(t),v(t)
(4.4,4.5)
Expected numbers W(O,t), V(O,t)
(4.23)
Unavailability Q(t)
(4.1)
(4.24,4.25)
Availability A(t)
Conditional intensity λ(t),μ(t)

Modified version of above flow chart will be discussed in the next chapter which we used in our program.

KITT - The code is an application of kinetic Tree theory and handles independent basic events which are non-repairable or repairable, provided they have constant failure fates and constant repair rates

B. Flow Chart for KITT Computations


f_1, g_1 for ıth component
ı = 1,.......,n
$w_1(t)$, $v_1(t)$, $w_1(0,t)$, $V_1(o,t)$, $Q_1(t)$, $_i(t)$,
$\mu_1(t)$ are obtaıned from above flow chart
(4.32)
(4.30a)
Cuts of w*(t)
Cutset Q*(t)
(4.38)
(7.31)
$w_s^{(1)}(t)$
Cutset λ *(t)
(4.36a)
(4.36a)
$Q_s(t)$
Inclusıon-exclusıon prıncıple
$w_B(t)$
Q_s max, Q_s mın
(4.40)
$w^{(2)}(t)$
$w_s(t)$, w_s max, w_s mın
$\lambda_s(t)$, λ_s max, λ_s mın
(4.40a,4.40b)

Exact tıme-dependent relıabılıty parameters are determined for each basıc events and cutse~~ts,~~ but for the system as a whole the parameters are obtaıned by upper or lower bound approxımatıons or by brackctıng. In bracketıng procedure, the varıous upper and lower bounds can be obtained as close to each other as desıred, and thus the exact value for system parameters are obtained if the user so choses.

w(t), v(t) are obtained by equations (4.21) and (4.22). For this purpose, numerical integration is used. W(0,t), V(0,t) are obtained by another integration for given t. Q is found by equation (4.23).

Minimal Cutset Parameters:

(a) Unavailability - A cutset is occurring if all the basic events in the cutset are occurring

$$Q^*(t) = \prod_{j=1}^{n} Q_j(t) \qquad (4.30a)$$

(b) Conditions failure intensity and Unconditional failure intensity:

$$\lambda^*(t) = \frac{w^*(t)}{1 - Q^*(t)} \qquad (4.31)$$

where

$$w^*(t) = \sum_{j=1}^{n} w_j(t) \prod_{\substack{i=1 \\ i \neq j}}^{n} Q_i(t) \qquad (4.32)$$

(c) Similarly

$$v^*(t) = \sum_{j=1}^{n} v_j(t) \prod_{\substack{i=1 \\ i \neq j}}^{n} [1 - Q_i(t)] \qquad (4.33)$$

$$\mu^*(t) = \frac{v^*(t)}{Q^*(t)} \qquad (4.34)$$

(d) Also

$$W^*(0,t) = \int_0^t w^*(u)\, du \qquad (4.35)$$

$$V^*(0,t) = \int_0^t v^*(u)\, du \qquad (4.36)$$

System Parameters·

(a) Unavailability

Let d_i = event that all the basic events of ith minimal cutset exist at time t.

The ith minimal cutset failure exist at time t.

By inclusion-exclusion principle

$$Q_s(t) = \Pr\left(\bigcup_{i=1}^{N_c} d_i \right) \quad ; \quad N_c = \text{No. of minimal cutsets}$$

$$= \sum_{i=1}^{N_c} \Pr(d_i) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \Pr(d_i \cap d_j(+ \dots$$

$$+ \dots + (-1)^m \sum_{1 \leq i_1 < i_2 \dots < i_m \leq N_c} \Pr(d_{i1} \cap d_{i2} \cap \dots \cap d_{im})$$

$$+ \dots + (-1)^{N_c - 1} \Pr(d_1 \cap d_2 \cap \dots \cap d_{N_c})$$

The mth term is the contribution to $Q_s(t)$ from m minimal cut set failures existing simultaneously at time t

$$Q_s(t) = \sum_{i=1}^{N_c} Q_i * (t) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} {}_{i,j}^{\pi} Q(t) + \dots$$

$$+ (-1)^{m-1} \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq N_c} {}_{i_1 \dots i_m}^{\pi} Q(t) + \dots$$

$$+ (-1)^{N_c - 1} \quad {}_{i_1 \dots \cancel{i}_{N_c}}^{\pi} Q(t) \qquad (4.36a)$$

where ${}_{i_1 \dots i_m}$ is the product of Q(t)'s for the basic events in cutset i_1 or $i_2 \dots$ or i_m.

Bracketing: If in above expression, only first term is taken, it is upper bound for $Q_s(t)$. If first two terms taken, it is lower bound for $Q_s(t)$. If 1st three terms taken, it is better upper bound for $Q_s(t)$ and so on.

Upper bound alternative expression is due to Esary and Proschan $= 1 - \prod_{i=1}^{N_c} [1 - Q_i *(t)]$.

This upper bound estimate is sometimes conservative estimate, but it is exact when the cutsets are disjoint sets of basic events.

(b) Unconditional failure intensity

$w_s(t)$ = Expected number of times the top event occurs at time t, per unit time

$w(t;1,\ldots m)$ = The unconditional failure intensity for a mode of failure which has as its basic failures the basic failures which are common members to all the mode failures $1,\ldots,m$

$$w_s^{(1)}(t) = \sum_{i=1}^{N_c} w_i *(t) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} w(t;i,j) \prod_{i,j} Q(t) + \sum_{i=3}^{N_c} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} w(t;i,j,k) \prod_{i,j,k} Q(t) + \ldots + (-1)^{N_c-1} w(t;1,\ldots,N_c) \prod_{1,\ldots N_c} Q(t) \quad (4.38)$$

A = The event that one or more of cutset failure occur at time t.

$B = \bar{A}$

$w_B(t;1,\ldots,m)\, dt = \Pr(e_1 \cap \ldots \cap e_m \cap B)$

$$= \sum_{i=1}^{N_c} \Pr(e_1 \cap \ldots \cap e_m \cap d_i) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \Pr(e_1 \cap \ldots \cap e_m \cap \ldots \cap d_i \cap d_j) \ldots + (-1)^{N_c-1} \Pr(e_1 \cap \ldots \cap e_m \cap d_1 \cap \ldots \cap d_{N_c})$$

$$w_s^{(2)} = \sum_{i=1}^{N_c} w_B(t;i)dt - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} w_B(t;i,j) + \ldots.$$

$$+ (-1)^{N_c-1} w_B(t;\ 1,\ldots,N_c) \qquad (4.40)$$

Bounds:

(i) $w_s(t)_{min} = w_s^{(1)}(t)_{min} - w_s^{(2)}(t)_{max}$

$w_s(t)_{max} = w_s^{(1)}(t)_{max} - w_s^{(2)}(t)_{min}$

(ii) if N_c is even:

$$Q_s(t) = Q_s(t)_{min}$$
$$w_s(t) = w_s(t)_{min}$$
$$\lambda_s(t) = \lambda_s(t)_{min}$$

If N_c is odd:

$$Q_s(t) = Q_s(t)_{max}$$
$$w_s(t) = w_s(t)_{max}$$
$$\lambda_s(t) = \lambda_s(t)_{max}$$

(iii) $\lambda_s(t)_{max} = \dfrac{w_{s,max}}{1-Q_{s,max}} \qquad (4.40a)$

$\lambda_s(t)_{min} = \dfrac{w_{s,min}}{1-Q_{s,min}} \qquad (4.40b)$

(c) Integrated number of failures

$$W_s(0,t) = \int_0^t w_s(u)\ du \qquad (4.41)$$

4.7 Alternative Formulation of System Analysis - Short-Cut Technique

J.B. Fussell formulated this technique which is basically back-of the-envelope guesstimate. It requires as input failure and repair rates for basic events and minimal cutsets. It assumes exponential distribution of and   and independent of component failures.

(a) Component Level Analysis:

$$Q_i = 1 - e^{-\lambda_i t} = \lambda_i t \qquad (4.42)$$

if $\lambda_i t < 0.1$

If components repairable

$$Q_i = \frac{\lambda_i}{\lambda_i + \mu_i} \left(1 - e^{-(\lambda_i + \mu_i)t}\right)$$

as t becomes large and if $\frac{\lambda_i}{\mu_i} << 0.1$

$$Q_i \cong \frac{\lambda_i}{\lambda_i + \mu_i} \cong \frac{\lambda_i}{\mu_i} \text{ if } t > \frac{2}{\mu_i} \qquad (4.43)$$

(b) Cutset level Analysis:

$$Q_i *(t) = \prod_{i=1}^{n} Q_i \qquad (4.44)$$

Using equations (4.32) and (4.24)

$$w_i * (t) \cong \sum_{j=1}^{n} [1 - Q_j(t)]\, \lambda_i(t) \prod_{\substack{i=1 \\ i \neq j}}^{n} Q_j(t)$$

Substituting equation(4.44) here,

$$w_i * (t) \cong Q_i * (t) \sum_{j=1}^{n} \frac{\lambda_j}{Q_j}(t) \qquad (4.45)$$

noting $1 - Q_j(t) \cong 1$

$$\lambda * (t) = \frac{w_i *(t)}{1 - Q_i *(t)} \qquad (4.46)$$

(c) <u>System Level Analysis using bounding Procedures</u>:

$$Q_s(t) \cong \sum_{i=1}^{N_c} Q_i *(t) \qquad (4.47)$$

$$\lambda_s \cong \sum_{i=1}^{N_c} \lambda_i * \qquad (4.48)$$

$$w_s(t) \cong \sum_{i=1}^{N_c} w_i * (t) \qquad (4.49)$$

or

$$\lambda_s \cong \frac{w_s(t)}{1-Q_s(t)} \; . \qquad (4.50)$$

CHAPTER 5

PROGRAM BOR SYSTEM QUANTIFICATION : MODKITT

5.1 Formulation for m Out of n Identical Systems

(a) Unavailability:

m out of n system: Top event unavailable if any of m or more systems out of n systems are unavailable.

By binomial distribution,

$$Q_s(t) = \sum_{k=m}^{n} \binom{n}{k} Q^k (1-Q)^{n-k} \qquad (5.1)$$

(b) Unconditional and conditional intensities for m out of n systems. There are $\binom{n}{m}$, mth order cutsets; $\binom{n}{m+1}$, (m+1)th order cutsets and so on. If we neglect higher order cutsets we are left with $\binom{n}{m}$, mth order cutsets. For finding minimal cutsets, we must do so.

Using equation (4.32) for any cutset of mth order

$$w^*(t) = m \; Q^{m-1} \; w$$

Since there will be m identical terms with valve w Q^{m-1} Finally using Fussell's approximation [13] equation (4.49)

$$w_s(t) = \binom{n}{m} m \; Q^{m-1} \; w$$

$$= \frac{n!}{m! \; n-m!} \; m \; Q^{m-1} \; w$$

$$w_s(t) = \frac{n!}{m-1! \; n-m!} \; Q^{m-1} \; w \qquad (5.2)$$

$$\lambda_s(t) = \frac{w_s(t)}{1-Q_s(t)} \qquad (5.3)$$

(c) Unconditional and conditional repair intensities using eqn.(4.33)

$$v*(t) = m\, v(t).(1-Q)^{m-1} \text{ for any of } \binom{n}{m}$$

cutsets of mth order.

Using Fussell's approximation

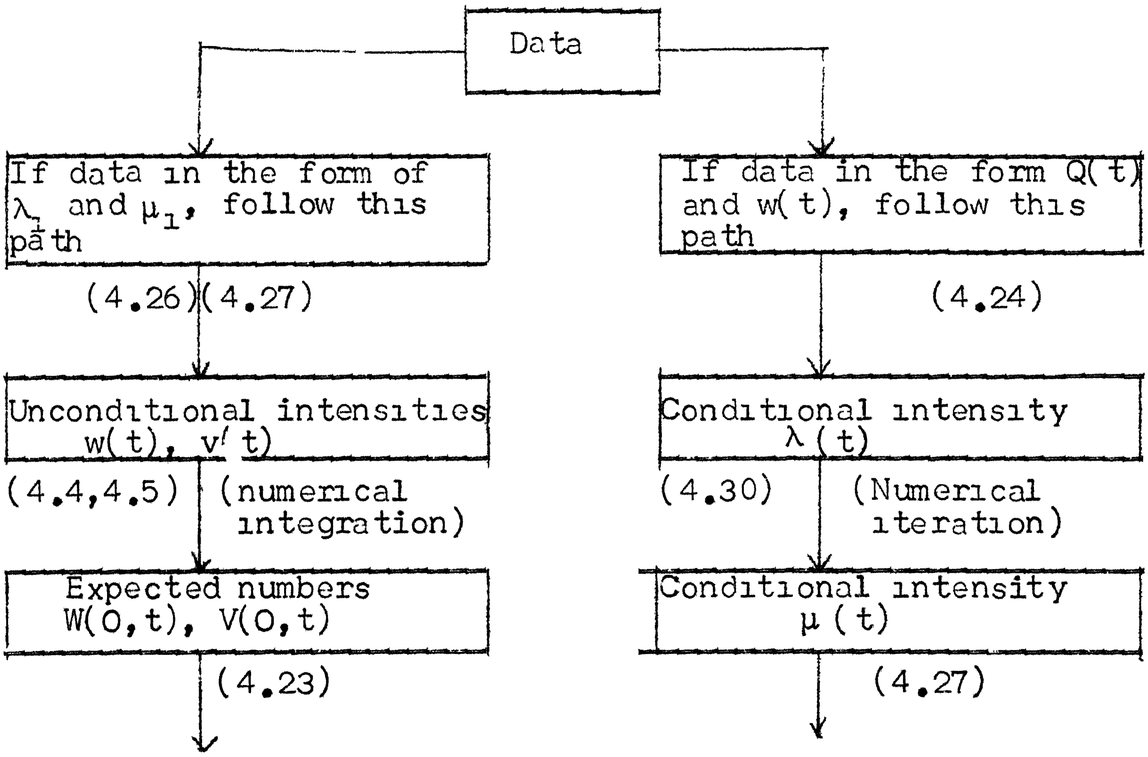
$$v_s(t) = \binom{n}{m} m\, v\, (1-Q)^{m-1}$$

$$= \frac{n!}{m-1!\ n-m!}\, (1-Q)^{m-1}.v \qquad (5.4)$$

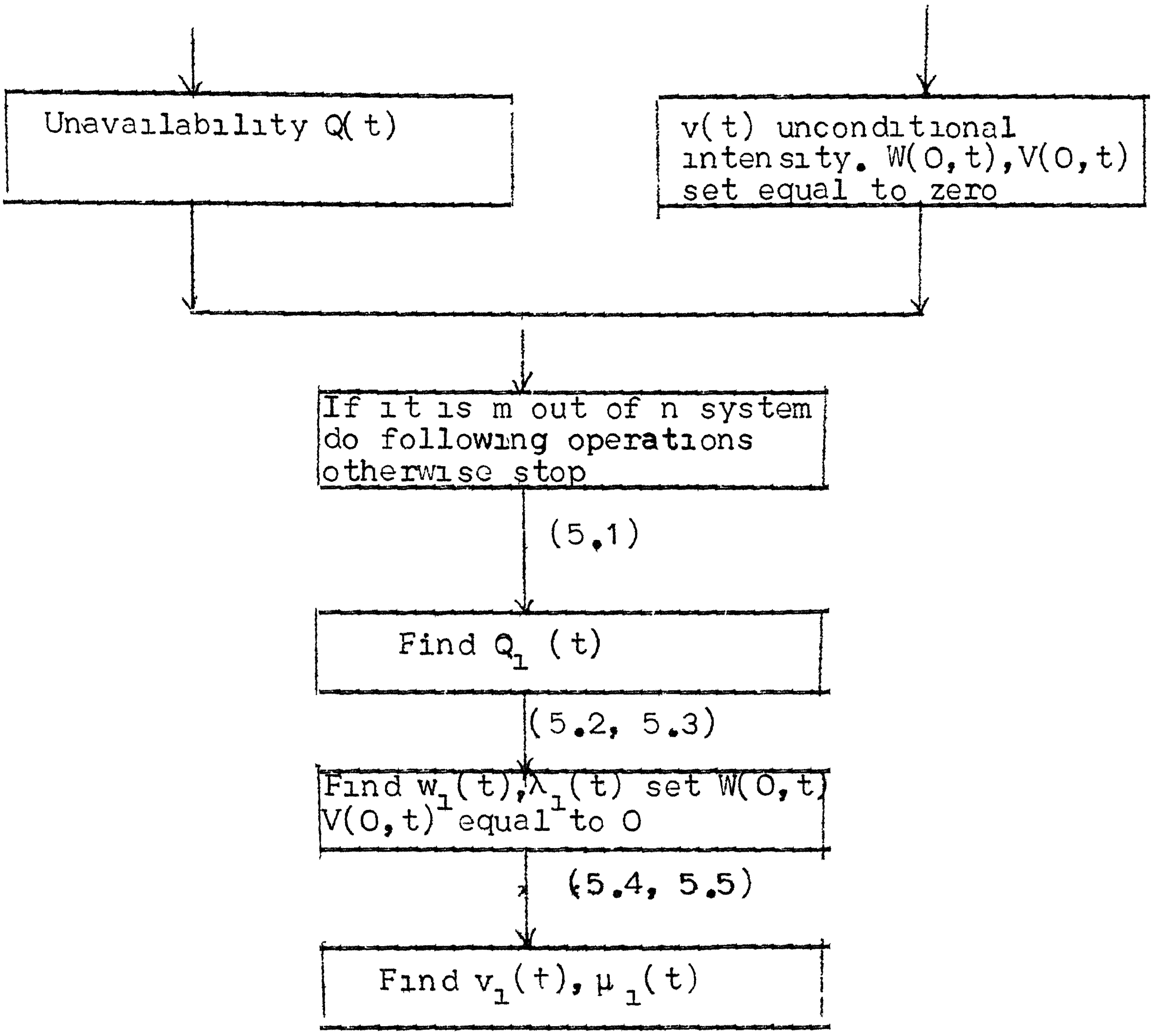
$$\mu_s(t) = \frac{v_s(t)}{Q_s(t)} \qquad (5.5)$$

5.2 Flow Chart for System Analysis

Before looking at final flow charge, let's have a look on the component or basic event analysis.



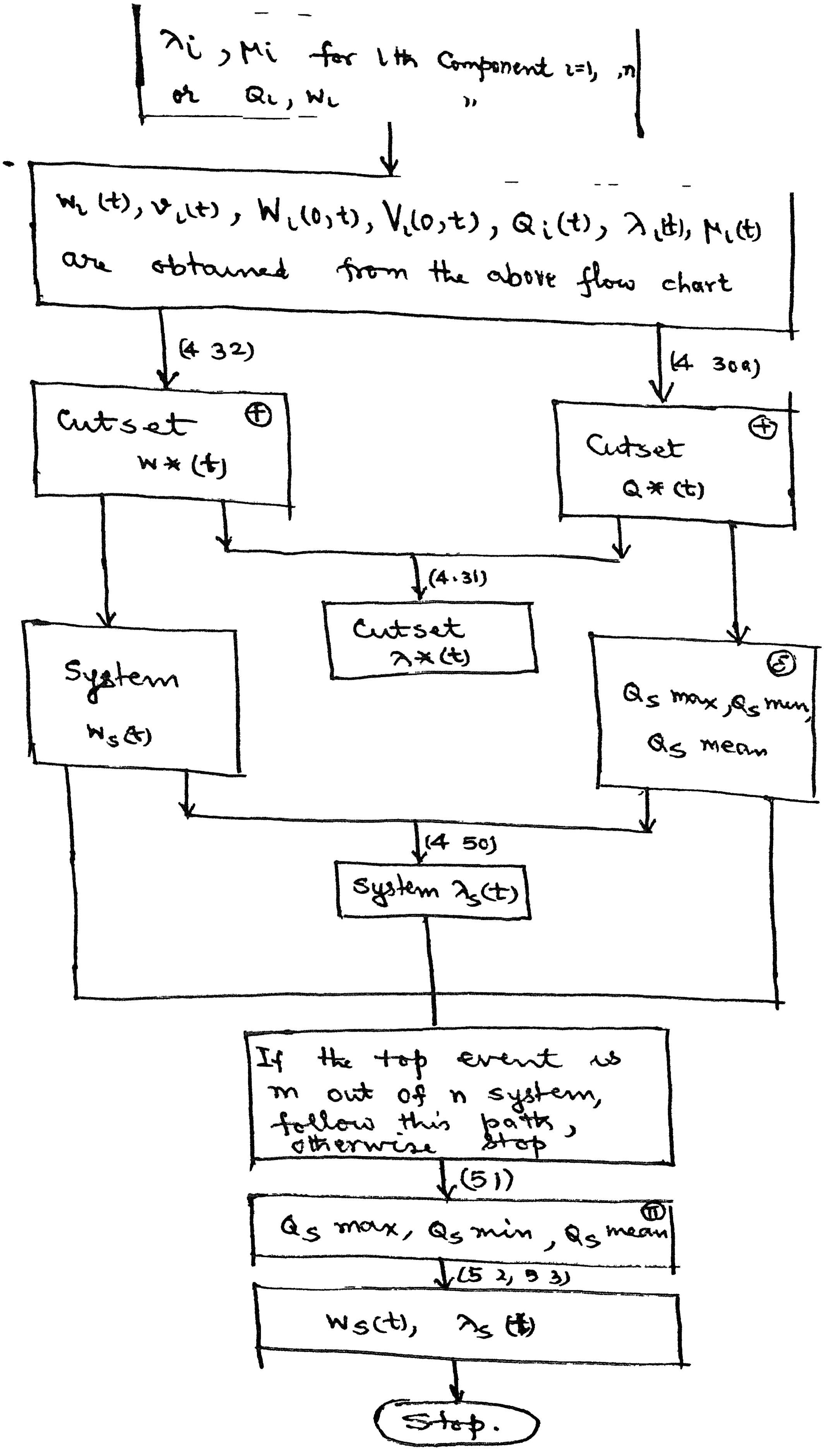
Data
If data in the form of λ_1 and μ_1, follow this path
If data in the form Q(t) and w(t), follow this path
(4.26)(4.27)
(4.24)
Unconditional intensities w(t), v(t)
Conditional intensity λ(t)
(4.4,4.5)
(numerical integration)
(4.30)
(Numerical iteration)
Expected numbers W(0,t), V(0,t)
Conditional intensity μ(t)
(4.23)
(4.27)


Unavailability Q(t)
v(t) unconditional intensity. W(O, t), V(O, t) set equal to zero
If it is m out of n system do following operations otherwise stop
(5.1)
Find Q_1 (t)
(5.2, 5.3)
Find w_1(t), λ_1(t) set W(O, t) V(O, t) equal to O
(5.4, 5.5)
Find v_1(t), μ_1(t)

5.3 A Note on m Out of n System

If we do not define m out of n system as done in section 5.1 rather as — "Top event available if any of m or more systems out of n systems are available" - we can still use the formulation of (5.1) if we do the following. Replace m by n-m+1.

System analysis:


λi, μi for ith Component i=1, ,n
or Qi, Wi "
wi(t), vi(t), Wi(0,t), Vi(0,t), Qi(t), λi(t), μi(t)
are obtained from the above flow chart
(4 32)
(4 30a)
Cutset W*(t)
Cutset Q*(t)
(4.31)
Cutset λ*(t)
System Ws(t)
Qs max, Qs min, Qs mean
(4 50)
System λs(t)
If the top event is m out of n system, follow this path, otherwise stop
(5 1)
Qs max, Qs min, Qs mean
(5 2, 5 3)
Ws(t), λs(t)
Stop.

⊕ Not all cutsets are taken. There are 3 options:

Option 0 => Take all cutsets

Option 1 => Maximum order of cutsets to be considered are specified. Higher order cutset $Q\{s\}$ are set to zero.

Option 2 Or more => % accuracy desired is specified.

If $\frac{Q^*_{max} - Q^*}{Q^*_{max}}$ > % assigned

Q^* is set to zero.

δ – Bracketting has been used to find $Q_{s,max}$, $Q_{s,min}$. If the first three terms taken in equation (4.36a), it gives $Q_{s,max}$. If first two terms taken, eqn.(4.36a) gives $Q_{s,min}$.

$$Q_s \text{ mean} = \sqrt{Q_{s,max} * Q_{s,min}} \qquad (5.6)$$

This geometric mean has been suggested in WASH-1400 report [4,5].

π – For all these quantities $Q_{s,max}$, $Q_{s,min}$, $Q_{s,mean}$, same equation (5.1) has been used. Hence $Q_{s,max}$ and $Q_{s,min}$ are not very accurate but $Q_{s,mean}$ is exact.

5.4 Input to Program

In file named FOR 24.DAT give:

In the first line - Time and decimal accuracy of result wanted.

In 2nd line onwards component no, λ (Q), μ (w), M,N, MN for all basic events. M,N indicates M out of N system.

MN $= 0 \Rightarrow$ P1 $= \lambda$ P2 $= \mu$

MN $\neq 0 \Rightarrow$ P1 $=$ Q(T) P2 $=$ w(T)

Next line must begin with 0 to indicate end of basic events.

Next line we have to give OPTION = ?

If Option = 0 $\Rightarrow$ take all cutsets.

If option = 1 $\Rightarrow$ Next line must contain MAX ORDER = ?

If option $\geq$ 2 => Next line must contain % = ?

Next lines contain cutsets one in each line till all cutsets are over. End of cutset information again indicated by a line starting with 0 or a black line.

Last line has M and N for the TOP GATE.

5.5 Output

First few lines have w(t), v(t), W(0,t), V(0,t), Lambda, Meu, and Q(t) for basic events.

Next we have

w * (t), λ * (t) and Q * (t) for the relevant cutsets only.

Lastly, λ_s, w(s), Q(s)min, Q(s)max, Q(s)mean for the total system.

Note: W(0,t), V(0,t) is set to zero for some basic events, this has been done to reduce computational effort. These quantities are required for calculation of Q(t), so if Q(t) is obtained by any alternate formulation or as such provided in input these quantities have no relevance and hence have been set to zero.

5.6 Merits of the Program

(i) In all formulation the most important quantity Q(t) (unavailability) calculation has been done exactly except for bracketing. First three terms of equation (4.36a) has been taken for upper bound which make it more close to actual value. Even for M out of N system unavailability calculation is exact.

(ii) Fussell's approximation has been used for quantities which are not of primary interest like w(t) of m out of n system and $w_s(t)$ for the entire system. Rest of the other quantities' formulation is exact like W(0,t), V(0,t), w(t), v(t) for components.

(iii) Answer accurate to any decimal place can be obtained. The iteration procedure and integration procedure has been written in a such a manner that it automatically reduces the step size of iteration for obtaining the desired decimal place accuracy.

(iv) Only those cutsets have been considered whose reliability is large enough according to certain criterion enumerated in the form of OPTION. This reduces the computational effort drastically. This reduces the computational effort drastically. In the current input, onlv 12 out of total 84 cutsets are important for 99% accuracy.

5.7 Demerits and Suggestions for Further Improvement

(i) The bracketting procedure has not been kept flexible. The number of terms for upper bound approximation and lower bound approximation has not been kept variables but arc 3 and 2 in this program. So desired accuracy of Q_{mean} can not be specified. This is not a particularly severe demerit since the successive terms of equation (4.36a) decrease by Q(t) which is a small number and hence two bounds found above is almost close to the actual value . In any case Q_{mean} is found by a method suggested by WASH - 1400 report.Hence,it is very dependable.

(ii) Fussell's approximation has been used for some quantities which is valid only for large time. This again is not a severe constraint since the approximation has been used only for quantities of secondary importance. MTTF and MTTR are in a few hours generally less than 100 whereas the time to be considered for calculation is in years. Hence condition of the kind equation (4.43) and large time is usually satisfied.

(iii) $v(t), \mu(t)$ for cutsets and overall system has not been found. This has not been done in KITT either. The reason for this is that these quantities are not of primary interest. Secondly, not all components are repairable hence calculation in those cases is not necessary.

5.8 Discussion of Result

For LMFBR primary loop, we obtained the following result for unavailability of the svstem fault tree we started out with.

Time	$\Lambda(s)$	w(s)	Q(s)min	Q(s) max	Q(s) mean
1 year	$0.5916\text{x}10^{-11}$	$0.5916\text{x}10^{-11}$	$0.25666\text{x}10^{-8}$	0.257x 10^{-8}	0.2569x 10^{-8}
40 years	$0.9374\text{x}10^{-10}$	$0.9374\text{x}10^{-10}$	$0.7857\text{x}10^{-7}$	0.7938x 10^{-7}	0.7898x 10^{-7}

After some improvement in the system by increasing redundancies in components (in parallel) as in the print-out, we got the following:

Time	λ (s)	w(s)	Q(s)min	Q(s) max	Q(s) mean
1 year	$0.8637x10^{-13}$	$0.8637x10^{-13}$	$0.38851x10^{-10}$	$0.38868x 10^{-10}$	$0.38859x 10^{-10}$
40 years	$0.3425x10^{-10}$	$0.3425y10^{-10}$	$0.24798x10^{-10}$	$0.25016x 10^{-7}$	$0.24907x 10^{-7}$

We observe the following facts from our analysis:

(i) With time, system unavailability increases. Even at the end of 40 years unavailability is $0.25x10^{-7}$ to $0.8x10^{-7}$ which is quite satisfactory.

(ii) As we increase redundancies, unavailability gets reduced. Our analysis can, thus, help in system design. We can decide the redundancies (standby and parallel components) based on some top event failure probability according to certain unavailability allocation techniques. Adequacy of the design can also be ascertained.

(iii) Even for 99% accuracy, not all minimum cutsets are important. So, it is better to concentrate on the important minimum cutsets rather than all. Also with time the important minimum cutsets keep on changing. Some old important cutsets no longer remain that important and some new adds to the list of important ones.

(iv) As the % accuracy required is decreased, not many new cutsets are added. So we can assign less accuracy desired without much increase in error. Like, in place of 99%, we can assign 85%.

(v) Some components are more critical in determining the system unavailability. Change of redundancy of some component has much pronounced effect on the overall unavailability than others.

Our result is closely comparable to the value $4x10^{-7}$ per reactor year obtained by F.J. Baloh, N.W. Brown, J. Graham, A.M. Smith, P.P. Zemanick [8] for Clinch River Breeder reactor plant. The value is also consistent with the primary allocation (goal) set which was less than equal to $8x10^{-7}$. This number was allocated on the basis of 10 CFR 100 criterion near plant site boundary [8].

CHAPTER 6

ILLUSTRATIVE PROGRAM FOR MINIMAL PATH AND CUTSET FOR AN ELECTRICAL SYSTEM

6.1 Introduction

Electrical systems differ from the basic nuclear system in that its fault tree has a network graph like structure. There are many feedback paths, parallel paths and interconnections. This is not like the fault tree containing OR and AND gate and hence a separate formulation is necessary for its analysis.

In the technique usea in the program, we do not require all the minimal paths to be deduced and checked. A few paths (called basic minimal paths) are deduced from the minimal path tree. The combination of failures that breaks the set of basic minimal paths is sufficient to deduce all the minimal cutsets of the n/w. The set of basic minimal paths is a subset of all the minimal paths of the n/w and the remaining minimal paths are not necessary for evaluating the minimal cutsets [20].

6.2 Definition

Minimal Path - It is a path from source to sink whereby no nodes are traversed more than once [19].

Basic Minimal Path - It is a minimal path in which no element of the path are linked to another by any branch in the n/w except with those elements before or after it in the path.

6.3 Program Details

Program is closely based on [20] algorithm details of which can be looked into. What we have done is an improvement in it to make it more efficient and doing minor corrections. The output format has been changed to make it more useful.

CHAPTER 7

FAULT TREE MODIFICATION PROGRAM (PROGRAM-4)

7.1 Object

Apart from AND and OR gates, there are many other gates which occur in the fault trees like XOR, NOT, NAND NOR, Priority AND, Inhibit etc. Priority AND and Inhibit gates are treated as AND gates. There should be some method to take care of XOR, NOT, NAND, NOR, M out of N gate. 'M' out of 'N' gate we took care at quantification level. The present program aims at taking care of XOR, NAND, NOR and NOT gates.

7.2 Difficulty

The gate equivalence for XOR leads to change in tree-level. Some method must be evolved to find sub-branch leading to change of level. Whole fault tree data must be modified accordingly. There are many zeros in some of basic event entries, these also must be taken care of.

7.3 Approach

XOR - gate has the following gate equivalence

$$A \oplus B = \bar{A}.B + A.\bar{B} \quad [13] \qquad \text{(See Figure 1)}$$

NOT - gates are taken care of by following -

De Morgan's law -

(i) $\overline{A + B} = \bar{A} . \bar{B}$

(ii) $\overline{A.B} = \bar{A} + \bar{B}$

Example

See Figure 2

XOR gate is first replaced by combination of two AND and one OR gate. Negation at any gate is removed by changing AND to OR or OR to AND gate with input negated. This process continued till we come to negated basic events and tree containing only AND and OR gates.

7.4 Program Details

As earlier we denote AND gate by an even number and OR gate by an odd Number. XOR gate is indicated by an odd Number larger than 100. NAND, NOR are indicated by a negative even or odd integer respectively. Next, we follow the following procedure:

(i) First entry of each basic event input data is checked for XOR gate. If no XOR gate, we go to next basic event. Then the second entry so on.

(ii) If any basic event has XOR gate, level of the branch containing that basic event is found. Level of branch is the number of gates encountered when we traverse from top event to that basic event by minimal path. If level of the branch is equal to level of tree, Level (L) of the tree is increased by one.

(iii) All the data right to the XOR gate is shifted by 1 with event number placed in array GN (I, LB + 1), where LB = level of the branch concerned.

(iv) GN (I, J) = 1 and GN (I, J+1) = 2 with GN(I,J+2) negated. 1 indicates OR gate and 2 AND gate.

NOTE: GN(I,J) = Gate no. for Ith basic event and at Jth level of the branch containing it.

(v) Next line consists of copy of GN(I,J) ∀ J with GN(I, J+1) = 4 and GN (I,J+2) unnegated. Hence all the entries below Ith line is shifted down by one.

(vi) Next line is looked into. If it contains the same XOR gate at the same level from top event, we do the following:

(a) GN (I,J) = 1 and GN (I, J+1) = 2 with GN (I,J+2) unnegated.1 indicates OR gate and 2 and 4 AND gates.

(b) Next line copy of GN (I,J) ∀ J.GN (I+1, J) = 1 and GN (I+1, J+1) = 4 with GN (I, J+2) negated.

(vii) This procedure is followed for all the basic event entry. If XOR gate different or at different level in the branch concerned, we go to step (ii).

(viii) When all entries has been looked into a particular level from top event, we consider the gates at next level from top event.

(ix) After entire input file has been modified, we restructure the tree according to the highest level decided in considering entire input file. If level is L, basic event in each line is placed at GN(I, L+1) and other entries are filled with zeroes suitably.

(x) Negation of a particular line is considered next. If any gate is negative then even is changed to odd and odd to even (AND to OR gate and OR to AND gate) and next non-zero entry is negated till we encounter the basic event number. This procedure is repeated for all lines in the input file.

7.5 Output

We get a fault tree containing only AND and/OR gates Some of the basic events are negated and some unnegated. The resultant fault tree can be easily fed to minimal cutset finding program to obtain minimal cutsets. These cutsets information can be given to fault tree quantification program to obtain top event unavailability and other parameters. Our minimal cutset finding program PCOMCP is such that actual gate number is unimportant as long as an even number indicates AND and an odd no indicates OR gate. Thus repetition of 1 and 2 as gate numbers is modified fault tree does not create problems in further analysis.

Modified fault tree and actual tree data can be traced easily from the print out. Two examples has been shown in the print out.

FINAL WORDS

We have complete package of fault tree analysis for nuclear systems. These system can contain electrical system (e.g. scram signals). Only thing left is fault tree construction program. But normally manual construction is done. Once fault tree is available, it can be analysed for top event failure probability using our package. Tree can have all kinds of gates. Final analysis depends on availability of basic event failure data. Assumption of constant failure and maintenance rates may be irritating. General time dependent analysis could be carried out by suitably modifying the programs. But this kind of exercise is unrealistic, since there is dearth of failure data.

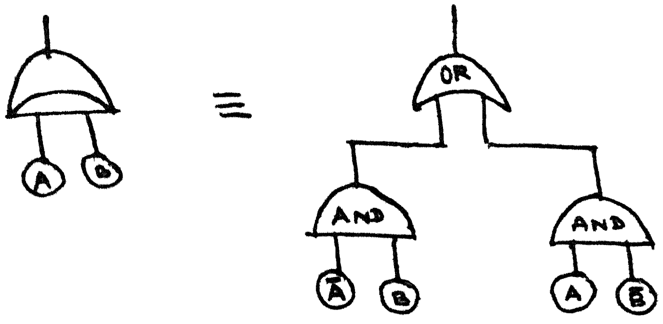
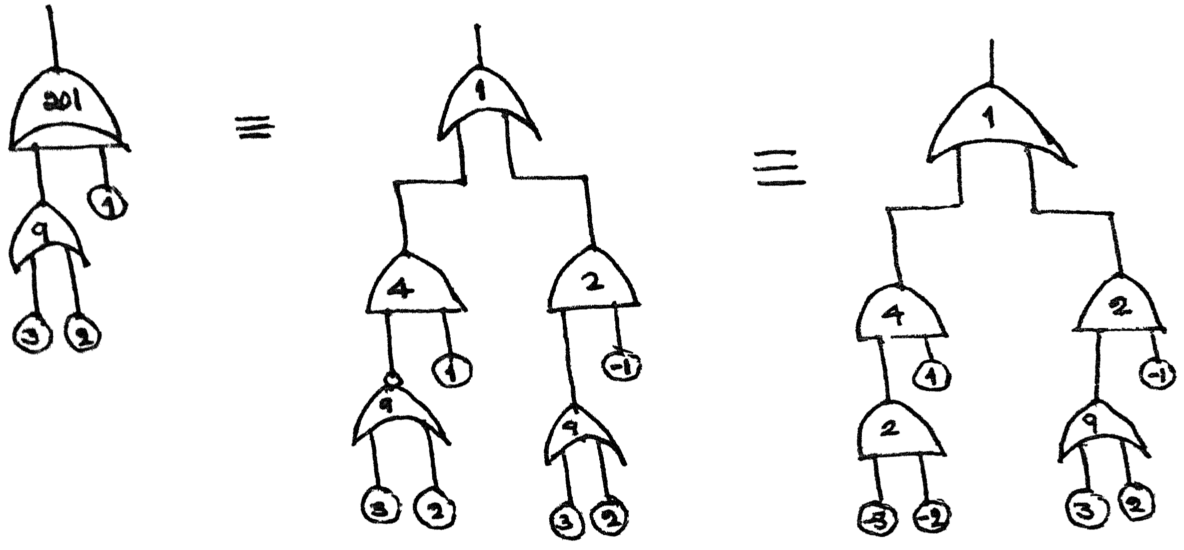


Figure 1 : Gate equivalence



gate equivalence

De Morgan's law

Figure 2 : Example and manual constouction.

APPENDIX - I

DATA TO BE USED

A. WASH-1400 [1, 2] report gives the following data:

(i) Pump

Mode	Q or λ		Median
(a) Failure to start	$3x10^{-4}$	$- 3x10^{-3}/d$	$1x10^{-3}/d$
(b) Failure to run given start in normal environment	$3x10^{-6}$	$- 3x10^{-4}/hr$	$3x10^{-5}/hr$

(ii) Motor Operated Valve

(a) Failure to operate	$3x10^{-4}$	$- 3x10^{-3}/d$	$1x10^{-3}/d$
(b) Failure to remain open	$3x10^{-5}$	$- 3x10^{-4}/d$	$1x10^{-4}/d$
(c) λ	$1x10^{-7}$	$- 1x10^{-6}/hr$	$3x10^{-7}/hr$
(d) Rupture λ	$1x10^{-9}$	$- 1x10^{-7}/hr$	$1x10^{-8}/hr$

(iii) Check Valve

(a) Failure to open	$3x10^{-5}$	$- 3x10^{-4}/d$	$1x10^{-4}/d$
(b) Internal Leakage	$1x10^{-7}$	$- 1x10^{-6}/hr$	$3x10^{-7}/hr$

(iv) Relief valve

(a) Failure to open	$3x10^{-6}$	$- 3x10^{-5}/d$	$1x10^{-5}/d$
(b) Premature open	$3x10^{-6}$	$- 3x10^{-5}/hr$	$1x10^{-5}/hr$

(v) Pipe

(a) _ 3" dia rupture	$3x10^{-11}$ - $3x10^{-8}$/hr	$1x10^{-9}$/hr
(b) _ 3" dia	$3x10^{-12}$ - $3x10^{-9}$/hr	$1x10^{-10}$/hr

(VI) MTTR for

Pump = 37 hr (typical)

Valve = 24 hr (typical)

B. Green and Bourne book gives the following data - [9]

(a) Pipe - $0.2x10^{-6}$ hr^{-1} [Failure rate]

(b) Control valve - $30x10^{-6}$ hr^{-1} [Failure rate]

(c) Solenoid valve - $30x10^{-6}$ hr^{-1} [Failure rate]

A.H. Earl has given the following information about pickering A and Bruce A life time incapacity upto the end of 1981 (CANDU) [16]:

	Pickering A	Bruce A
Heat transport pump	0.2 (years)	0.2 (years)
Pressure tubes	4.9 (years)	0.3 (years)
Boilers	0.5 (years)	2.4 (years)
Turbines & Generators	5.8 (years)	6.6 (years)
Heat exchangers	0.9 (years)	0.0 (years)
Valves	0.4 (years)	0.0 (years)

J.R. Aupied and H. Procaccia have given the following data [11] for valves:

System	No.of oper. hrs.	MTTR	Mean unavai. time	hr^{-1}	on demand failure
Pneumatic valves	700000	22	53	$16x10^{-6}$	$5x10^{-3}/d$
Check valve	600000	39	54	$8x10^{-6}$	$0.6x10^{-3}/d$
Large flow rate valves	350000	15	70	$50x10^{-6}$	$4.5x10^{-3}/d$
Small flow rate valves	330000	8	49	$54x10^{-6}$	$1.7x10^{-3}/d$

Steam valves

Safety relief valve (low pressure)	560000	34		$59x10^{-6}$	$2x10^{-3}/d$
Safety relief valve (high pressure)	$2.25x10^{6}$	24		$49x10^{-6}$	$8x10^{-3}/d$
Check valve	$1.43x10^{6}$	11		$6x10^{-6}$	$0.03x10^{-3}/d$
Motor operated valve	$1.425x10^{6}$	20		$0.7x10^{-6}$	$0.7x10^{-3}/d$
Pneumatic valve	340000	24		$65x10^{-6}$	$1x10^{-3}/d$

From A.H. Earl's data [ ]

for pump

unavailability $= \frac{.2}{25} = 8x10^{-3}$ at the end of one year.

$Q = 1 - e^{-\lambda t} \Rightarrow \lambda = \frac{1}{t} \ln \frac{1}{1-Q}$

$\therefore \lambda = 9.171x10^{-7}$

for Heat exchangers

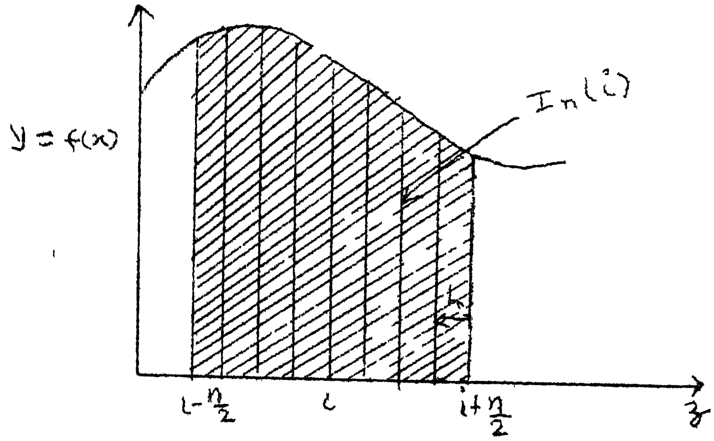
$Q = \frac{.9}{25}$

$\lambda = 4.185x10^{-6}$

APPENDIX II

ALGORITHM OF NUMERICAL INTEGRATION [15, 16]

Simpson's Rule [15]



$$I_{2n}(i) = \frac{h}{3}\left(f_i + 4f_{i+1} + f_{i+2}\right) \qquad (II.1)$$

$$\text{Error} = -\frac{h^5}{90}\, f^{IV} \qquad (II.2)$$

$$S = \sum_{i=1,3,5,..,n-2} \frac{h}{3}\left(f_i + 4f_{i+1} + f_{i+2}\right) \qquad (II.3)$$

$$= \frac{h}{3}\left(f_1 + 4f_2 + 2f_3 + 4f_4 + 2f_5 + \ldots . + f_{n+1}\right) \qquad (II.4)$$

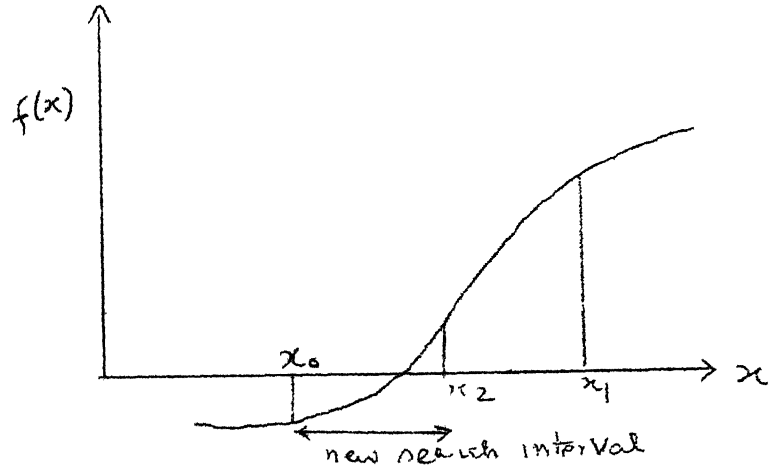
Hence f is to be tabulated at odd number of points.

If accuracy upto Nth decimal place is desired $e = 0.5 * 10^{-N}$.

APPENDIX III

ITERATION TECHNIQUE

Bisection Method [15]



We begin by picking two trial points which enclose the root this is indicated by $f(x_0)$ and $f(x_1)$ being of opposite sign. The interval (x_0, x_1) is bisected and mid point denoted by x_2, i.e. $x_2 = (x_1 + x_0)/2$. If $f(x_2) = 0$ then x_2 is the root. If $f(x_2) > 0$ then root is between x_0 and x_2. Hence replace x_1 by x_2 and search for root in this half interval.

If $f(x_2) < 0$ then root is between x_2 and x_1. Hence replace x_0 by x_2 and again bisect the interval.

This bisection procedure is repeated till search interval is smaller than the precision with which answer is wanted.

Note that this method always encloses the root in the search interval and search interval is halved each time. Thus in 10 iterations, the search interval reduces by $2^{10} \simeq 1000$ and in 20 by $10^{20} \simeq 10^6$.

REFERENCES

1. WASH 1400: Reactor Safety Study, U.S.,N.R.C., 1975, Appendix I.

2. WASH 1400. Reactor Safety Study, U.S., N.R.C., 1975, Appendix II.

3. "FETA - A Fast Fault Tree Analysis Program", T.Matsuoka, Nucl. Engg. and Design, Vol. 91, Jan. 1986, No.1, pp. 93-101.

4. "Efficient construction of Minimal Cutsets from Fault Trees" - S. Garrebba, P. Mussio, F. Naldi, G. Reina, G. Volta, IEEE Trans. on Reliability, Vol. R-26, No.2, June 1977, pp. 88-93.

5. "FATRAM - A core efficient cutset algorithm", D.M. Rasmuson, N.H. Marshall, IEEE Trans. on Reliability, Vol. R-27, No.4, Oct. 1978, pp. 250-253.

6. "AFTP - Fault Tree Analysis Program", R.A. Pullen, IEEE Trans. on Reliability, Vol. R-33, No.2, June 1984, pp. 171.

7. "The Heat Transfer System of SNR-300", S. Dreyer, Nucl. Engg. International, Vol. 21, No. 246, July 1976, pp. 49-50.

8. "Clinch River Breeder-Reactor Plant System Reliabilty", F.J. Baloh, N.W. Brown, J. Graham, A.M. Smith, P.P. Zemanick, IEEE Trans. on Reliability, Vol. R-25, No.3, Aug. 1976, pp. 191-196.

9. "Reliability Technology" - A.E. Green and A.J. Bourne, Wiley - Interscience, 1972.

10. "Reliability of CANDU Heat Transport Pump", A.H. Earl, Special Issue on the International Symposium on Reliability of Reactor Pressure Component, Vol. 81-82, pp. 85, March 1983.

11. "SRDF: A System for Collecting Reliability data from French PWR Power Plants. Method of Failure Analysis. Application to the Processing of Valve Data", J.R.Aupied and H. Procaccia, Vol. 81, No.1, 1984, Aug.(II), pp.127-137.

12. "Code of Federal Regulation, Title 10, "Energy"", Jan. 1, 1975.

13. "Reliability Engineering and Risk Assessment", E.J. Henley and H. Kumamoto, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1981.

14. "Reliability Engineering", A K. Govil, Tata McGraw-Hill Publishing Company Ltd., New Delhi, 1983.

15. "Computer Oriented Numerical Methods" - V. Rajaraman, Prentice Hall of India Private Ltd., New Delhi, 2nd Ed., 1984.

16. "Numerical Methods in FORTRAN" - J.M. McCormick, M.G. Salvadori, Prentice Hall of India Private Ltd., New Delhi, 1968.

17. "Approaches to Companying Cutset Enumeration Algorithms", IEEE Trans. on Reliability, Vol. R-28, 1979 April, pp. 62-63, by A. Rosenthall.

18. P.C. Williams, A.A. Esan, R.N. Adams, "Efficient Computation of Distribution System Reliability", Proc. 6th Power System Computation, Conference, Damstadd, 1978.

19. M.O. Locks, "Relationship between Minimal Path Sets and Cutsets", IEEE Trans. on Reliability, Vol. R-27, 1978, June, p. 106.

20. "A New Technique in Minimal Path and Cutset Evaluation" - G.B. Jasmon, O.S. Kai, IEEE Trans. on Reliability, Vol. R-24, No.2, June, 1985.

```
ATCON version 102(2067) running C sequence 5995 in stream 1
nput from DSKC:C.CTL[15100,150064]
tput to  DSKC:C.LOG[15100,150064]
b parameters
me:00:01:00   Core:100P   Unique:YES   Restart:YES   Output:NOLOG

LOGIN 15100/150064 /DEFER/SPOOL:ALL/TIME:50/CORE:100P/LOCATE:10/NAME:"MANOJ KUM
OB 16 I I T KANPUR 603A(3) TTY116
GNJSP Other jobs same PPN:18]
408    13-Dec-86        Sat
u have some mail. Type *.BOX.

TY C.FOR
4:09:01]
C**        Program for Minimal Cutsets using Prime Numbers

C          PROGRAM NAME "PCOMCP"

C          THIS PROGRAM USES PRIME NO. CODING FOR BASIC EVENTS.
C          AFTER MINIMIZATION DECODES THEM BACK.
C          INPUT LINES AS MANY AS NO. OF BASIC EVENTS ONLY.
C          UPTO 100 BASIC INPUTS AND 100 OR GATES COULD BE ANALYSED.
C          METHOD OF PROVIDING INPUT:=
C          IN THE FIRST LINE GIVE LEVEL NO. ,NO. OF OR GATES & MAX
C          ORDER OF MINIMAL CUTSET WANTED.
C          LEVEL NO. IS MAXM. NO. OF GATES ENCOUNTERED IN COMING
C          FROM TOP GATE TO ANY OF THE BASIC EVENTS.
C          FROM NEXT LINE ONWARDS GIVE GATE NOS. ENCOUNTERED IN
C          COMING FROM TOP TO ANY BASIC EVENT.
C          LAST NO. OF THESE LINES IS BASIC EVENT NO.
C          LAST LINE MUST BEGIN WITH 0 AND ARBITRARY OTHER NOS.
C          ORDER OF BASIC EVENTS DOES NOT MATTER.
C          OR GATES MUST HAVE ODD GATE NO. AND AND GATE EVEN NO.
C          INPUTS ARE TO BE GIVEN IN FILE NAME 'FOR20.DAT'.
C          OUTPUT COMES IN FILE NAME'FOR21.DAT'.

C          ------------------------------------------------------------
C          VARIABLE INDEX:=
C          ----------------
C          P      - PRIME NOS.
C          EN     - EVENT NOS.
C          LP     - LEVEL POINTER
C          ST     - STORED VALUE
C          OR     - NO. OF OR GATES: NO.>THIS COULD ALSO BE GIVEN
C          PF     - PRIME FACTOR
C          DIV    - DIVIDER FOR FINDING PRIME NOS.
C          S      - STORAGE
C          K      - INTERMEDIATE STORAGE
C          L      - LEVEL
C          MO     - MAX ORDER UP TO WHICH CUTSETS ARE WANTED
C          P1F(100)        -PRIME FACTOR ARRAY
C          NG(100)         -GATE NO            ,,
C          N(100,100)      -OPERATION          ,,
C          P(100)          -PRIME NOS.         ,,
C          PF(100)         -PRIME FACTOR       ,,
C          S(100)          -STORAGE            ,,
C          ------------------------------------------------------------

           INTEGER P,C,EV,LP,ST,OR,PF,DIV,S,P1F
           DIMENSION NG(100),N(100,100),P(100),PF(100),S(100).P1F(100)
           READ(20,1),L,OR,MO
```

```
 1        FORMAT(1X,3(I3,1X))
          P(1)=2
 C        NN IS CHECK NO.
          NN=2
          I=1
 4        IF(I.GT.99)GO TO 2
          NN=NN+1
          DIV=2
 5        IF(MOD(NN,DIV).EQ.0) GO TO 4
          DIV=DIV+1
          IF(DIV.NE.NN) GO TO 5
          I=I+1
 C        PRIME NOS. STORED IN P(I)
          P(I)=NN
          GO TO 4
 2        CONTINUE
          DO 20 I=1,100
          NG(I)=0
          PF(I)=0
          P1F(I)=0
          DO 21 J=1,100
          N(I,J)=0
 21       CONTINUE
 20       CONTINUE
 C        INITIALISATION
 15       READ(20,11) ((NG(I),I=1,L),EN)
 11       FORMAT(1X,100(I3,1X))
          DO 7 I=1,100
          S(I)=0
 7        CONTINUE
          IF(NG(1).EQ.0)GOTO 170
 C        CONDITION INDICATES END OF DATA
          DO 12 I=1,100
          IF(EN.NE.I) GO TO 12
          EN=P(I)
          GO TO 13
 12       CONTINUE
 C        LOOP FOR PRIME NO CODING
 13       LP=L
          DO 30 I=1,L
          IF(NG(LP).NE.0) GO TO 35
          LP=LP-1
 30       CONTINUE
 C        ASSIGNMENT OF INITIAL LEVEL POINTER i.e. MAX NO. OF GATE
 C                        FROM TOP
          GO TO 40
 36       DO 38 J=1,OR
          IF(N(LP,J).NE.0) GO TO 40
 38       CONTINUE
          N(LP,1)=EN
          GO TO 45
 40       IF(MOD(NG(LP),2).NE.0)GO TO 55
 C        IF NO. EVEN IT IS AND GATE ;IF ODD IT IS OR GATE
          DO 56 J=1,OR
          N(LP,J)=N(LP,J)*EN
 56       CONTINUE
 C        MULTIPLY (IF AND GATE) ALL THE ELEMENTS IN ARRAY
          GO TO 80
 55       DO 50 J=1,OR
          IF(N(LP,J).NE.0) GO TO 50
          GO TO 55
 50       CONTINUE
 55       N(LP,J)=EN
```

```
C       STORE IF OR GATE IN ARRAY
30      IF(LP.EQ.1) GO TO 45
        LP=LP-1
C       DECREASE LEVEL POINTER
        DO 85 J=1,OR
        IF(N(LP,I).EQ.0) GO TO 85
        GO TO 90
85      CONTINUE
        DO 95 I=1,OR
        N(LP,I)=N(LP+1,I)
        N(LP+1,I)=0
95      CONTINUE
C       SHIFT IF N(I,I) NOT VACANT TO N(I-1,I)
        GO TO 45
90      IF(MOD(NG(LP),2).EQ.0) GO TO 100
C       CHECK EVEN OR ODD
        DO 110 J=1,OR
        IF(N(LP,J).NE.0) GO TO 110
        GO TO 120
110     CONTINUE
120     K=0
        DO 130 I=J,100
        K=K+1
        IF(N(LP+1,K).EQ.0) GO TO 80
        N(LP,I)=N(LP+1,K)
        N(LP+1,K)=0
130     CONTINUE
C       SHIFTING OPERATION
        GO TO 80
100     DO 131 I=1,OR
        IF(N(LP,I).EQ.0)GOTO 135
        S(I)=N(LP,I)
        N(LP,I)=0
131     CONTINUE
135     K=0
        LOP=I-1
        DO 140 I=1,LOP
        DO 150 J=1,OR
        IF(K.GT.100)GOTO 160
        LOOP=S(I)*N(LP+1,J)
        IF(LOOP.EQ.0)GOTO 140
        K=K+1
        N(LP,K)=LOOP
150     CONTINUE
140     CONTINUE
        DO 145 J=1,OR
        N(LP+1,J)=0
145     CONTINUE
C       INITIALIZATION AFTER SHIFTING
160     GO TO 80
170     DO 180 I=1,100
        DO 190 J=1,100
        IF(I.EQ.J) GO TO 190
        IF(N(1,I).EQ.0)GOTO 180
        IF(N(1,J).EQ.0)GOTO 190
        IF((MOD(N(1,J),N(1,I))).EQ.0) N(1,J)=0
190     CONTINUE
180     CONTINUE
        WRITE(21,195)
195     FORMAT(1X,'Following are the Minimal Cut Sets')
        DO 300 I=1,100
        DO 800 JI=1,100
        PF(JI)=0
```

```
                   P1F(JI)=0
        C          INITIALIZATION
        300        CONTINUE
                   K=1
                   M=1
                   IF(N(1,I).EQ.0) GO TO 300
        220        IF(P(K).GT.N(1,I)) GO TO 290
                   IF(MOD(N(1,I),P(K)).EQ.0) GO TO 250
        270        K=K+1
        C          LEAVE OUT THOSE ELEMENTS IN N(I,I) WHICH ARE MULTIPLE
        C                   OF EACH OTHER
                   GO TO 220
        250        N(1,I)=N(1,I)/P(K)
        C          PRIME FACTORIZATION
                   DO 260 J=1,100
                   IF(P(J).NE.P(K)) GO TO 260
                   PF(M)=J
                   M=M+1
                   GOTO 220
        260        CONTINUE
                   GO TO 220
        290        DO 310 IN=1,99
                   DO 320 J=IN+1,100
                   IF(PF(J).EQ.PF(IN))PF(J)=0
        320        CONTINUE
        310        CONTINUE
                   K=0
                   DO 265 J=1,100
                   IF(PF(J).EQ.0) GO TO 265
                   K=K+1
                   P1F(K)=PF(J)
        C          P1F STORES THOSE FACTORS WHICH ARE NOT REPEATED
                   IF(J.EQ.100)GO TO 280
        265        CONTINUE
                   IF(MO.EQ.0)GOTO 280
                   IF(K.GT.MO)GOTO 300
        280        WRITE(21,321),(P1F(IL),IL=1,K)
        321        FORMAT(100I4)
        300        CONTINUE
                   STOP
                   END

.TY FOR20.DAT
[14:09:04]
   3    9
   1    2    0    2
   1    2    5    1
   1    2    5    3
   1    4    0    4
   1    4    3    5
   1    4    3    6
   1    5    0    7
   1    5    0    8
   0

.EX C.FOR
[14:09:04]
LINK:   Loading
[LNKXCT C execution]

STOP

END OF EXECUTION
```

```
CPU TIME: 0.39  ELAPSED TIME: 0.78
EXIT

.TY FOR21.DAT
[14:09:05]
 Following are the Minimal Cut Sets
    1    2    3
    4    5
    4    6
    7
    8

.KJOB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 16  User MANOJ KUMAR [15100,150064]
Logged-off TTY116  at 14:09:05  on 13-Dec-86
Runtime: 0:00:01, KCS:19, Connect time: 0:00:10
Disk Reads:301, Writes:11
BATCON version 102(2067) running C sequence 5998 in stream 1
Input from DSKC:C.CTL[15100,150064]
Output to  DSKC:C.LOG[15100,150064]
Job parameters
Time:00:01:00   Core:100P   Unique:YES   Restart:YES   Output:NOLOG

.LOGIN 15100/150064 /DEFER/SPOOL:ALL/TIME:50/CORE:100P/LOCATE:10/NAME:"MANOJ K
JOB 16 I I T KANPUR 603A(3) TTY116
[LGNJSP Other jobs same PPN:18]
1410    13-Dec-86       Sat
You have some mail. Type *.BOX.

.TY FOR20.DAT
[14:10:51]
    7   25    5
    2    1    3    5    7    0    0    1
    2    1    3    5    7    0    0    2
    2    1    3    5    9   11    0    3
    2    1    3    5    9   11    0    4
    2    1    3    5    9    0    0    5
    2    1    3   13   15   17    0    6
    2    1    3   13   15   17    0    7
    2    1    3   13   15   19   21    8
    2    1    3   13   15   19   21    9
    2    1    3   13   15   19    0   10
    2    1    3   13    0    0    0   11
    2    1   25   27   29    0    0   12
    2    1   25   27   29    0    0   13
    2    1   25   27   31   33    0   14
    2    1   25   27   31   33    0   15
    2    1   25   27   31   39    0   16
    2    1   25   27   31   39    0   17
    2    1   25   35   37    0    0   18
    2    1   25   35   37    0    0   19
    2    1   25   35   41    0    0   20
    2    1   25   35   41    0    0   21
    2   23    0    0    0    0    0   22
    2   23   43   45    0    0    0   23
    2   23   43   45    0    0    0   24
    2   23   43    0    0    0    0   25
    0

 EX C.FOR
```

```
[14:10:52]
LINK:   Loading
[LNKXCT C execution]

STOP

END OF EXECUTION
CPU TIME: 1.97  ELAPSED TIME: 2.66
EXIT

.TY FOR21.DAT
[14:10:55]
 Following are the Minimal Cut Sets
     1    22
     1    23
     1    24
     1    25
     2    22
     2    23
     2    24
     2    25
     3    22
     3    23
     3    24
     3    25
     4    22
     4    23
     4    24
     4    25
     5    22
     5    23
     5    24
     5    25
     6    22
     6    23
     6    24
     6    25
     7    22
     7    23
     7    24
     7    25
     8    22
     8    23
     8    24
     8    25
     9    22
     9    23
     9    24
     9    25
    10    22
    10    23
    10    24
    10    25
    11    22
    11    23
    11    24
    11    25
    12    22
    12    23
    12    24
    12    25
    13    22
    13    23
```

```
  13   24
  13   25
  14   22
  14   23
  14   24
  14   25
  15   22
  15   23
  15   24
  15   25
  16   22
  16   23
  16   24
  16   25
  17   22
  17   23
  17   24
  17   25
  18   22
  18   23
  18   24
  18   25
  19   22
  19   23
  19   24
  19   25
  20   22
  20   23
  20   24
  20   25
  21   22
  21   23
  21   24
  21   25

.KJOB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 16  User MANOJ KUMAR [15100,150064]
Logged-off TTY115  at 14:10:56  on 13-Dec-85
Runtime: 0:00:02, KCS:45, Connect time: 0:00:07
Disk Reads:235, Writes:11
```

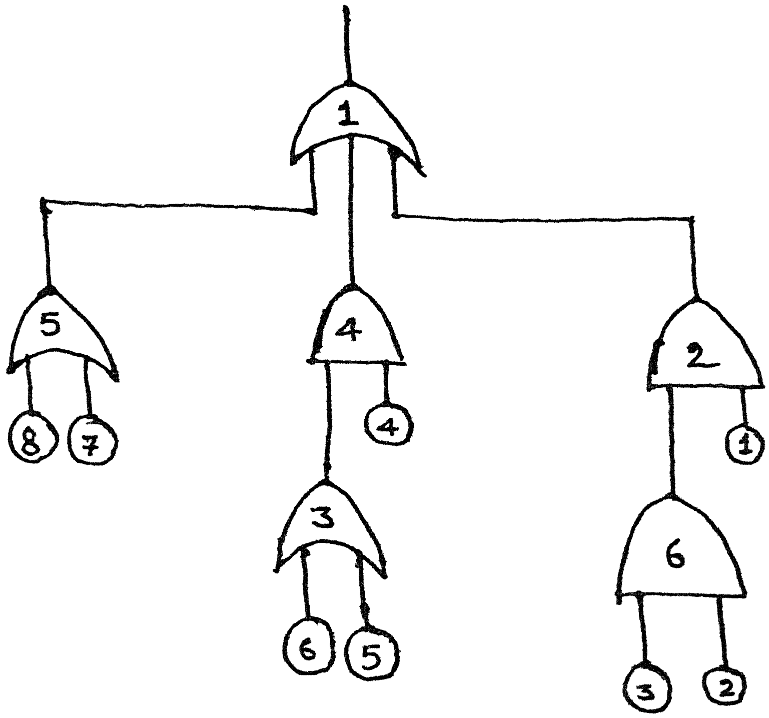


Figure · Example 1 ~~for~~ PCOMCP

```
                                                   1


BATCON version 102(2067) running A sequence 2752 in stream 1
Input from DSKC:A.CTL[15100,150064]
Output to  DSKC:A.LOG[15100,150064]
Job parameters
Time:00:01:00    Core:100P   Unique:YES    Restart:YES    Output:NOLOG

.LOGIN 15100/150064 /DEFER/SPOOL:ALL/TIME:50/CORE:100P/LOCATE:10/NAME:"MANOJ KUM
JOB 23 I I T KANPUR 603A(3) TTY115
[LGNJSP Other jobs same PPN:27]
1340     29-Nov-86          Sat

.TY A.FOR
[13:40:55]
00100
00200   C           PROGRAM NAME "MODKITT"
00300   C
00400   C           THIS PROGRAM USES KITT FORMULATION FOR FINDING SYSTEM UNAVAI-
00500   C           LIBILITY AND SHORT CUT CALCULATION METHOD GIVEN BY J. FUSSELL
00600   C           FOR FINDING SYSTEM PARAMETERS LIKE W(S) AND LAMBDA(S).W(0,t)
00700   C           AND V(0,t) ARE ACCURATELY FOUND FOR BASIC EVENTS.
00800   C              IN FINDING SYSTEM Q(S) BRACKETING PROCEDURE HAS BEEN USED.
00900   C           FIRST BRACKET GIVES Q(S)MIN AND SECOND Q(S)MAX.
01000   C           FOR BASIC EVENTS WE CAN SPECIFY M OUT OF N SYSTEM.IN THE PRO-
01100   C           GRAM WE HAVE ASSUMED THE N SYSTEM IDENTICAL.SIMILARLY WE CAN
01200   C           SPECIFY LAST GATE TO BE M OUT OF N.FOR INTERMEDIATE GATES WE
01300   C           WILL MODIFY THE FAULT TREE TO TAKE INTO ACC. ANY M OUT OF N
01400   C           GATE.
01500   C              THERE ARE THREE OPTION TO THE USER.FIRST IS WHEN ALL CUTSETS
01600   C           ARE TO BE TAKEN INTO ACC.SECOND IS WHEN MAX. ORDER OF CUTSETS
01700   C           ARE TO BE TAKEN INTO ACC.THIRD IS WHEN MAX. PERCENTAGE OF THE
01800   C           MAX OF ALL CUTSET Q IS SPECIFIED.
01900   C             -:METHOD OF INPUTTING THESE INFORMATION:-
02000   C             -----------------------------------------
02100   C           IN FILE NAME FOR24.DAT GIVE
02200   C           IN THE FIRST LINE TIME AND DECIMAL ACCURACY OF W(0.t) AND V(0,t)
02300   C           WANTED.
02400   C           IN SECOND LINE COMP NO,LAMBDA,MEU,M,N,MN FOR ALL BASIC EVENTS.
02500   C           M AND N ARE AS MENTIONED ABOVE.WHEN ALL THE BASIC EVENTS ARE
02600   C           OVER LAST LINE MUST BE EITHER A BLANCK LINE OR A LINE STARTING
02700   C           WITH 0
02800   C           MN=0 => P1=LAMBDA ;P2=MEU (DATA TO BE GIVEN)
02900   C           MN  NOT=0 => P1=Q(T)  P2=W(T)  (DATA TO BE GIVEN)
03000   C           NEXT LINE HAS OPTION=? OPTION=0 =>TAKE ALL CUTSETS.
03100   C           OPTION=1 =>TAKE MAX ORDER SPECIFICATION.
03200   C           OPTION=2 OR MORE =>MAX % IS ASSIGNED.
03300   C           IF OPTION=1 NEXT LINE CONTAINS MAX ORDER=?
03400   C           IF OPTION=2 OR MORE IT CONTAINS %=?
03500   C           NEXT LINES HAVE MINIMAL CUTSET INFORMATION.
03600   C           THESE AGAIN MUST END WITH EITHER A BLANCK LINE OR A LINE
03700   C           STARTING WITH 0
03800   C           LAST LINE HAS M AND N OF TOP GATE.
03900   C
04000   C           OUTPUT HAS FIRST w(t),v(t),W(0,t),V(0,t),LAMBDA,MEU AND Q(t)
04100   C           OF BASIC COMPONENTS.
04200   C           NEXT IT HAS w(t),Lamoda,Q(t) OF VARIOUS RELEVENT CUTSETS.
04300   C           LAST IT HAS LAMBDA,w(s),Q(S)MIN Q(S)MAX AND MEAN VALUE OF TOTAL
04400   C           SYSTEM.MEAN IS CALCULATED BY SQRT(MAX*MIN).
04500   C           NOTE: WHEN P1 IS PROVIDED AS Q(T) & P2 AS w(T) ;W &V (INTEGRATED
04600   C           W &V) IS ASSIGNED =0 ALSO  WHEN M OUT OF N SYSTEM CONSIDERED....
04700   C           [illegible]
04800   C           VARIABLE INDEX:=


                                                   2


04900   C           -------------
05000   C           LL- LOWER LIMIT OF UNAVAILABILITY
05100   C           UL- UPPER LIMIT OF Q
05200   C           CN- COMPONENT NO.
05300   C           LAMBM- MEAN LAMBDA
05400   C           WM- MEAN W
```

```
C*    COUNT- COUNTER
C*    FACT- FACTORIAL
C*    CS(100,100)- CUTSET MATRIX
C*    CQ(100)- CUTSET Q
C*    CW(100)- CUTSET W
C*    KS(100)- CUTSET ORDER ARRAY
C*    CLAMB(100)- CUTSET LAMBDA
C*    SV(100)- v(t)
C*    F- Q BY MARKOV'S FORMULATION
C*    F1- w(t)        ,,
C*    F2- v(t)        ,,
C*    P1- LAMBDA OR Q AS INDICATED EARLIER
C*    P2- MEW OR w(T)             ,,
C*    T- TIME
C*    N1- DECIMAL PLACE UPTO WHICH ACCURACY WANTED
C*    M,N- M OUT OF N SYSTEM
C*    MN- INDICATOR WHETHER P1 IS LAMBDA OR Q; P2 IS MEU OR W
C*             DEPENDING ON WHETHER IT IS 0 OR NONZERO
C*    S= STORAGE FOR Q
C*

      REAL LL,LAMBDA,MEU,LAMBM
      INTEGER CN,CS,OPTION,COUNT,FACT
      DIMENSION Q(100),SW(100),CS(100,100),CQ(100),CW(100),KS(100),
     1CLAMB(100),SV(100)
      EXTERNAL F,F1,F2,FACT
      COMMON P1,P2
      READ(24,20)T,N1
   20 FORMAT(F8.1,1X,I2)
      WRITE(5,25)
   25 FORMAT(/,1X,'COMP NO    t           w(t)            v(t)
     1   w(0,t)          V(0,t)           LAMBDA
     2MEU              Q(t)  ')
      DO 100 I=1,100
      READ(24,30)CN,P1,P2,M,N,MN
   30 FORMAT(1X,I3,2(1X,E10.3),3(1X,I2))
      IF(CN.EQ.0)GOTO 105
      IF(MN.NE.0)GOTO 35
      SW(CN)=F1(P1,P2,T)
      SV(CN)=F2(P1,P2,T)
      CALL INT(F1,0.,T,N1,W)
      CALL INT(F2,0.,T,N1,V)
      Q(CN)=W-V
      GOTO 38
   36 Q(CN)=P1
      SW(CN)=P2
      LAMBDA=SW(CN)/(1.-Q(CN))
      P1=LAMBDA
      P2=0.
      CALL  ET(F,Q(CN),N1)
      MEU=P2
      SV(CN)=F2(P1,P2,T)
   38 IF(M.EQ.0)GOTO 70
      ST=Q(CN)
      Q(CN)=0.
      N2=N
      DO 40 J=M,N
```

```
      J2=J
      IV=FACT(N2)/(FACT(J2)*FACT(N2-U2))
      Q(CN)=Q(CN)+FLOAT(IV)*(ST**J)*((1.-ST)**(N2-J))
   40 CONTINUE
      RT=FLOAT(FACT(N)/(FACT(M-1)*FACT(N-M)))
      SW(CN)=RT*(ST**(M-1))*SW(CN)
      SV(CN)=RT*((1.-ST)**(M-1))*SV(CN)
      MEU=SV(CN)/Q(CN)
      LAMBDA=SW(CN)/(1.-Q(CN))
      W=0.
      V=0.
      GOTO 80
   70 LAMBDA=SW(CN)/(1.-Q(CN))
      MEU=SV(CN)/Q(CN)
   80 WRITE(5,90),(CN,T,SW(I),SV(I),W,V,LAMBDA,MEU,Q(I))
   90 FORMAT(2X,I3,1X,F8.1,7(1X,E13.6))
  100 CONTINUE
  105 READ(24,110)OPTION
  110 FORMAT(7X,I1)
      IF(OPTION.EQ.0)GOTO 145
      IF(OPTION.GE.2)GOTO 130
      READ(24,120)MO
```

```
 120  FORMAT(10X,I2)
      PC=0.
      GOTO 143
 130  READ(24,140)PC
 140  FORMAT(2X,F4.1)
 145  MO=100
 143  WRITE(5,146)
 146  FORMAT(//,1X,'CUT      t         w*(t)        Lambda*(t)         Q*(t)'
      COUNT=0
 150  DO 165 I=1,100
      READ(24,160),(CS(I,I1),I1=1,100)
 160  FORMAT(100(1X,I3))
      IF(CS(I,1).EQ.0)GOTO 170
      COUNT=COUNT+1
 165  CONTINUE
C     SIMULATION OF CUTSET Q,CUTSET W &CUTSET LAMBDA EQUATIONS.
 170  DO 250 I=1,COUNT
      KS(I)=0
      CQ(I)=1.
      CW(I)=0.
      DO 200 I1=1,100
      IF(CS(I,I1).EQ.0)GOTO 205
      KS(I)=KS(I)+1
 200  CONTINUE
 205  DO 230 I1=1,KS(I)
      J=CS(I,I1)
      CQ(I)=CQ(I)*Q(J)
      S=1.
      DO 220 L=1,KS(I)
      IF(J.EQ.CS(I,L))GOTO 220
      K=CS(I,L)
      S=S*Q(K)
 220  CONTINUE
      CW(I)=CW(I)+SW(J)*S
 230  CONTINUE
      CLAMB(I)=CW(I)/(1.-CQ(I))
 250  CONTINUE
C     AS PER OPTION ACTION IS TAKEN. IF OPTION=0,ALL CUTSETS TAKEN
C     INTO ACCOUNT. IF OPTION=1,IF CUTSET ORDER GREATER THAN MAX ORDER
C     SPECIFIED THEN Q IS REDUCED TO ZERO.
C     IF OPTION>=2 IF ACCURACY (MAX Q-Q)/MAX Q *100 >% ASSIGNED THEN
```

```
C     AGAIN Q REDUCED TO ZERO.
      IF(OPTION.EQ.0)GOTO 280
C     TO FIND MAX CQ(I)=CV
      CV=CQ(1)
      DO 260 I=2,COUNT
      IF(CQ(I).GT.CV)CV=CQ(I)
 260  CONTINUE
      DO 270 I=1,COUNT
      IF(OPTION.EQ.1)GOTO 262
      IF(((CV-CQ(I))*100./CV).LE.PC)GOTO 270
      GOTO 264
 262  IF(KS(I).LE.MO)GOTO 270
 264  CQ(I)=0.
      CLAMB(I)=0.
      CW(I)=0.
 270  CONTINUE
 280  DO 285 I=1,COUNT
      IF(CQ(I).EQ.0.0)GOTO 285
      WRITE(5,282),(I,T,CW(I),CLAMB(I),CQ(I))
 282  FORMAT(1X,I2,1X,F8.1,3(1X,E13.6))
 285  CONTINUE
C     SIMULATION OF SYSTEM Q MAX,Q MIN,Q MEAN,LAMBDA,w(T)
      UL=0.
C     LOOP FINDS FIRST TERM OF SYSTEM Q EQUATION.
      DO 290 I=1,COUNT
      UL=UL+CQ(I)
 290  CONTINUE
      COR1=0.
C     FINDS SECOND TERM OF SYSTEM Q EQUATION AS COR1.
      DO 340 I=2,COUNT
      DO 330 J=1,I-1
      CUM=CQ(I)
      DO 320 K1=1,KS(J)
      DO 310 K2=1,KS(I)
      IF(CS(I,K2).EQ.CS(J,K1))GOTO 320
 340  CONTINUE
      KL=CS(J,K1)
      CUM=CUM*Q(KL)
```

```
  320 CONTINUE
      COR1=COR1+CUM
  330 CONTINUE
  340 CONTINUE
      LL=UL-COR1
      COR2=0.
C     FINDS THIRD TERM OF SYSTEM Q EQUATION AS COR2.
      DO 400 I=3,COUNT
      DO 390 J=2,I-1
      DO 380 K=1,J-1
      CUM=CQ(I)
      DO 360 K1=1,KS(J)
      DO 350 K2=1,KS(I)
      IF(CS(J,K1).EQ.CS(I,K2))GOTO 360
  350 CONTINUE
      KL=CS(J,K1)
      CUM=CUM*Q(KL)
  360 CONTINUE
      DO 378 K0=1,KS(K)
      DO 370 K2=1,KS(I)
      IF(CS(K,K0).EQ.CS(I,K2))GOTO 378
  370 CONTINUE
      DO 375 K1=1,KS(J)
      IF(CS(K,K0).EQ.CS(J,K1))GOTO 378
  375 CONTINUE
```

```
      KL=CS(K,K0)
      CUM=CUM*Q(KL)
  378 CONTINUE
      COR2=COR2+CUM
  380 CONTINUE
  390 CONTINUE
  400 CONTINUE
      UL=LL+COR2
      QM=SQRT(UL*LL)
      READ(24,450)M,N
  450 FORMAT(2I3)
      IF(M.EQ.0)GOTO 465
C     AS PER OUR FORMULATION , TO FIND Q FOR M OUT OF N SYSTEM
      ST=QM
      QM=0.
      ST1=UL
      ST2=LL
      UL=0.
      LL=0.
      DO 460 J=M,N
      J2=J
      N2=N
      IV=FACT(N2)/(FACT(J2)*FACT(N2-J2))
      QM=QM+FLOAT(IV)*(ST**J2)*((1.-ST)**(N2-J2))
      UL=UL+FLOAT(IV)*(ST1**J2)*((1.-ST1)**(N2-J2))
      LL=LL+FLOAT(IV)*(ST2**J2)*((1.-ST2)**(N2-J2))
  460 CONTINUE
  465 WM=0.
      LAMBM=0.
C     FUSSEL'S APPROXIMATION
      DO 470 I=1,COUNT
      WM=WM+CW(I)
      LAMBM=LAMBM+CLAMB(I)
  470 CONTINUE
      IF(M.EQ.0)GOTO 475
      RT=FLOAT(FACT(N)/(FACT(M-1)*FACT(N-M)))
      WM=RT*(ST**(M-1))*WM/(1.-(ST**M))
      LAMBM=WM/(1.-QM)
  475 WRITE(5,480)
  480 FORMAT(//,1X,'      t          LAMBDA(S)          W(S)          Q(S)MIN
     1  Q(S)MAX        Q(S)MEAN   ')
      WRITE(5,490)(T,LAMBM,WM,LL,UL,QM)
  490 FORMAT(1X,F8.1,5(1X,E13.6))
      STOP
      END

C     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C**   NUMERICAL INTEGRATION
C     THIS  SUBROUTINE USES SIMPSON FORMULA
C     X1, X2 ARE LOWER AND UPPER LIMITS OF INTEGRATION
C     N IS DECIMAL PLACE UPTO WHICH ACCURACY OT INTE. WANTED
C     OUTPUTS ALSO STEP USED FOR  DESIRED ACCURACY
C     FUNCTION TO BE FED THROUGH A EXTERNAL FUNCTION
C     F1 DEFINES THE FUNCTION
```

```
        SUBROUTINE INT(F1,X1,X2,N,I1)
        REAL I0,I1
        COMMON P1,P2
        H=(X2-X1)/2.
        I=2
        S1=F1(P1,P2,X1)+F1(P1,P2,X2)
        S2=0.
        S4=F1(P1,P2,X1+H)
```

```
        I0=0.
        I1=(S1+4.*S4)*(H/3.)
        IF(I1.EQ.0.)GOTO 6
10      IF(ABS((I1-I0)/I1).LE.0.5*10.**(-N))GOTO 5
        S2=S2+S4
        S4=0.
        X=X1+(H/2.)
        DO 2 J=1,I
        S4=S4+F1(P1,P2,X)
        X=X+H
2       CONTINUE
        H=H/2.
        I=2*I
        I0=I1
        I1=(S1+2.*S2+4.*S4)*(H/3.)
        GOTO 10
5       CONTINUE
5       RETURN
        END

C       ********************************************************************
        FUNCTION F(P1,P2,T)
        IF(((P1+P2)*T).GT.60.)F=P1/(P1+P2)
        IF(((P1+P2)*T).LE.60.)F=(P1/(P1+P2))*(1.-EXP(-T*(P1+P2)))
        RETURN
        END
C       ***********************************************************************

        FUNCTION F1(P1,P2,T)
        IF(((P1+P2)*T).GT.60.)F1=P1*P2/(P1+P2)
        IF(((P1+P2)*T).LE.60.)F1=((P1*P2)+(P1**2)*EXP(-T*(P1+P2)))/(P1+P
        RETURN
        END
C       ************************************************************************

        FUNCTION F2(P1,P2,T)
        IF(((P1+P2)*T).GT.60.)F2=P1*P2/(P1+P2)
        IF(((P1+P2)*T).LE.60.)F2=(P1*P2/(P1+P2))*(1.-EXP(-T*(P1+P2)))
        RETURN
        END
C       ************************************************************************

        INTEGER FUNCTION FACT(N)
        FACT=1
        IF(N.EQ.0)GOTO 1000
        DO 999 I=1,N
        FACT=FACT*I
999     CONTINUE
1000    RETURN
        END
C       !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

        SUBROUTINE ET(F,Q,N)
        COMMON P1,P2
        Y1=1.
        Y0=0.
1050    IF(ABS((Y1-Y0)/Y1).LE.0.5*10.**(-N))GOTO 1500
        P2=(Y1+Y0)/2.
        Z=F(P1,P2,T)
        IF(Q-F(P1,P2,T))1300,1500,1400
1300    Y1=P2
        GOTO 1700
1400    Y0=P2
```

```
36400    1700      GOTO 1050
36500    1500      RETURN
36600              END

.TY FOR24.DAT
[13:41:00]
350400.0   6
   1  +1.000E-10  +5.000E-09
   2   1.000E-12   4.000E-11
   3  +1.000E-10  +5.000E-09
   4  +1.000E-12  +4.000E-11
   5  +1.500E-05  +1.000E-04     3    5
   6  +1.000E-10  +5.000E-09
   7  +1.000E-12  +4.000E-11
   8  +1.000E-10  +5.000E-09
   9  +1.000E-12  +4.000E-11
  10  +9.170E-07   1.000E-05     3    4
  11   4.184E-06   2.000E-05     2    3
  12   1.000E-10   5.000E-09
  13   1.000E-12   4.000E-11
  14   1.000E-10   5.000E-09
  15   1.000E-12   4.000E-11
  16   2.500E-06   2.500E-05     1    3
  17   3.000E-05   3.000E-04     2    4
  18   1.000E-10   5.000E-09
  19   1.000E-12   4.000E-11
  20   8.000E-06   5.000E-05     2    3
  21   8.000E-06   5.000E-05     2    3
  22   3.000E-06   5.000E-04
  23   2.000E-09   1.200E-07
  24   2.000E-10   1.000E-08
  25   2.500E-06   2.500E-04
   0
 TION=7
 70.0
   1   22
   1   23
   1   24
   1   25
   2   22
   2   23
   2   24
   2   25
   3   22
   3   23
   3   24
   3   25
   4   22
   4   23
   4   24
   4   25
   5   22
   5   23
   5   24
   5   25
   6   22
   6   23
   6   24
   6   25
   7   22
   7   23
   7   24
   7   25
```

```
   8   22
   8   23
   8   24
   8   25
   9   22
   9   23
   9   24
   9   25
  10   22
  10   23
  10   24
  10   25
  11   22
  11  23
  11  24
  11  25
  12  22
  12  23
  12  24
  12  25
  13  22
  13  23
  13  24
  13  25
  14  22
  14  23
  14  24
  14  25
  15  22
  15  23
  15  24
  15  25
  16  22
  16  23
  16  24
  16  25
  17  22
  17  23
  17  24
  17  25
  18  22
  18  23
  18  24
  18  25
  19  22
  19  23
  19  24
  19  25
  20  22
  20  23
  20  24
  20  25
  21  22
  21  23
  21  24
  21  25
   0
  3   3

.EX A.F/R
[13:41  1]
LINK:   Loading
[LNKXCT  A execution]
```

```
                                              9

COMP ID    t          W(t)           V(t)          W(0,t)         V(0,t)          LAMBDA           MEU           Q(t
   1 350400.0  0.999965E-10  0.175044E-12  0.350394E-04  0.306769E-07  0.100000E-09  0.500002E-08  0.350087E-
   2 350400.0  0.100000E-11  0.140216E-15  0.350400E-06  0.245587E-11  0.100000E-11  0.400164E-10  0.350397E-
   3 350400.0  0.999965E-10  0.175044E-12  0.350394E-04  0.306769E-07  0.100000E-09  0.500002E-08  0.350087E-
   4 350400.0  0.100000E-11  0.140216E-15  0.350400E-06  0.245587E-11  0.100000E-11  0.400164E-10  0.350397E-
   5 350400.0  0.787257E-05  0.307515E-03  0.000000E+00  0.000000E+00  0.804236E-05  0.145657E-01  0.211123E-
   6 350400.0  0.999965E-10  0.175044E-12  0.350394E-04  0.306769E-07  0.100000E-09  0.500002E-08  0.350087E-
   7 350400.0  0.100000E-11  0.140216E-15  0.350400E-06  0.245587E-11  0.100000E-11  0.400164E-10  0.350397E-
   8 350400.0  0.999965E-10  0.175044E-12  0.350394E-04  0.306769E-07  0.100000E-09  0.500002E-08  0.350087E-
   9 350400.0  0.100000E-11  0.140216E-15  0.350400E-06  0.245587E-11  0.100000E-11  0.400164E-10  0.350397E-
  10 350400.0  0.681855E-07  0.830613E-05  0.000000E+00  0.000000E+00  0.633278E-07  0.398929E-02  0.208211E-
  11 350400.0  0.359118E-05  0.171662E-04  0.000000E+00  0.000000E+00  0.390094E-05  0.216182E-03  0.794065E-
  12 350400.0  0.999965E-10  0.175044E-12  0.350394E-04  0.306769E-07  0.100000E-09  0.500002E-08  0.350087E-
  13 350400.0  0.100000E-11  0.140216E-15  0.350400E-06  0.245587E-11  0.100000E-11  0.400164E-10  0.350397E-
```

```
14 350400.0   0.999965E-10   0.175044E-12   0.350394E-04   0.306769E-07   0.100000E-09   0.500002E-08   0.350387E-
15 350400.0   0.100000E-11   0.140216E-15   0.350400E-06   0.245587E-11   0.100000E-11   0.400164E-10   0.350397E-
16 350400.0   0.681823E-05   0.681774E-05   0.000000E+00   0.000000E+00   0.907488E-05   0.274167E-04   0.248571E+
17 350400.0   0.297521E-04   0.297521E-03   0.000000E+00   0.000000E+00   0.311143E-04   0.679563E-02   0.437312E-
18 350400.0   0.999965E-10   0.175044E-12   0.350394E-04   0.306769E-07   0.100000E-09   0.500002E-08   0.350387E-
19 350400.0   0.100000E-11   0.140216E-15   0.350400E-06   0.245587E-11   0.100000E-11   0.400164E-10   0.350397E-
20 350400.0   0.570749E-05   0.356718E-04   0.000000E+00   0.000000E+00   0.601946E-05   0.688290E-03   0.518267E-
21 350400.0   0.570749E-05   0.356718E-04   0.000000E+00   0.000000E+00   0.601946E-05   0.688290E-03   0.518267E-
22 350400.0   0.298211E-05   0.298211E-05   0.404497E+01   0.103900E+01   0.300001E-05   0.499679E-03   0.696305E-
23 350400.0   0.199863E-08   0.823239E-10   0.700558E-03   0.145259E-04   0.200000E-08   0.120000E-06   0.686032E-
24 350400.0   0.199986E-09   0.699550E-12   0.700775E-04   0.122534E-06   0.200000E-09   0.100000E-07   0.699549E-
25 350400.0   0.247525E-05   0.247525E-05   0.867425E+00   0.857524E+00   0.250000E-05   0.249997E-03   0.990111E-


CUT        t          W*(t)        Lambda*(t)        Q*(t)
44 350400.0   0.232107E-06   0.232290E-06   0.785213E-03
61 350400.0   0.782254E-06   0.783417E-06   0.448408E-02
64 350400.0   0.683029E-06   0.584715E-06   0.245212E-02


     t           LAMBDA(S)         W(S)          Q(S)MIN        Q(S)MAX        Q(S)MEAN
350400.0   0.937377E-10   0.937377E-10   0.785703E-07   0.793917E-07   0.789799E-07
STOP

END OF EXECUTION
CPU TIME: 28.65 ELAPSED TIME: 2:38.02
EXIT

.KJOB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 23  User MANOJ KUMAR [15100,150064]
Logged-off TTY115  at 13:43:45  on 29-Nov-86
Runtime: 0:00:30, KCS:648, Connect time: 0:02:54
Disk Reads:285, Writes:3
BATCON version 102(2067) running A sequence 2755 in stream 1
Input from DSKC:A.CTL[15100,150064]
Output to  DSKC:A.LOG[15100,150064]
Job parameters
Time:00:01:00   Core:100P   Unique:YES   Restart:YES   Output:NOLOG

.LOGIN 15100/150064 /DEFER/SPOOL:ALL/TIME:60/CORE:100P/LOCATE:10/NAME:"MANOJ KUMAR"
JOB 23 I I T KANPUR 603A(3) TTY115
[LGNJSP Other jobs same PPN:27]
1346    29-Nov-86        Sat

.TY FOR24.DAT
```

```
[13:46:24]
  8760.0   6
   1  +1.000E-10  +5.000E-09
   2   1.000E-12   4.000E-11
   3  +1.000E-10  +5.000E-09
   4  +1.000E-12  +4.000E-11
   5  +1.500E-05  +1.000E-04   3   5
   6  +1.000E-10  +5.000E-09
   7  +1.000E-12  +4.000E-11
   8  +1.000E-10  +5.000E-09
   9  +1.000E-12  +4.000E-11
  10  +9.170E-07   1.000E-05   3   4
  11   4.184E-06   2.000E-05   2   3
  12   1.000E-10   5.000E-09
  13   1.000E-12   4.000E-11
  14   1.000E-10   5.000E-09
  15   1.000E-12   4.000E-11
  16   2.500E-06   2.500E-05   1   3
  17   3.000E-05   3.000E-04   2   4
  18   1.000E-10   5.000E-09
  19   1.000E-12   4.000E-11
  20   8.000E-06   5.000E-05    2    3
  21   8.000E-06   5.000E-05    2    3
  22   3.000E-06   5.000E-04
  23   2.000E-09   1.200E-07
  24   2.000E-10   1.000E-08
  25   2.500E-06   2.500E-04
   0
OPTION=7
%=70.0
   1    22
   1    23
   1    24
   1    25
   2    22
   2    23
   2    24
   2    25
   3    22
   3    23
   3    24
   3    25
   4    22
   4    23
   4    24
   4    25
   5    22
   5    23
   5    24
   5    25
   6    22
   6    23
   6    24
   6    25
   7    22
   7    23
   7    24
   7    25
   8    22
   8    23
   8    24
   8    25
   9    22
```

```
   9    23
   9    24
   9    25
  10    22
  10    23
  10    24
  10    25
  11    22
  11    23
  11    24
  11    25
  12    22
  12    23
  12    24
  12    25
  13    22
  13    23
  13    24
  13    25
  14    22
  14    23
  14    24
  14    25
  15    22
  15    23
  15    24
  15    25
  16    22
  16    23
  16    24
  16    25
  17    22
  17    23
  17    24
  17    25
```

```
18   22
18   23
18   24
18   25
19   22
19   23
19   24
19   25
20   22
20   23
20   24
20   25
21   22
21   23
21   24
21   25
  0
 3   3

.EX A.FOR
[13:46:25]
LINK:    Loading
[LNKXCT A execution]

COMP NO    t          w(t)          v(t)          W(0,t)        V(0,t)        LAMBDA        MEU           Q(t
   1    8760.0   0.999999E-10  0.438050E-14  0.876000E-06  0.191373E-10  0.100000E-09  0.500069E-08  0.875980E-
   2    8760.0   0.100000E-11  0.356174E-13  0.876000E-08  0.156580E-14  0.100000E-11  0.406592E-10  0.876000E-
   3    8760.0   0.999999E-10  0.438050E-14  0.876000E-06  0.191373E-10  0.100000E-09  0.500069E-08  0.875980E-
```

12

```
   4    8760.0   0.100000E-11  0.356174E-13  0.876000E-08  0.156580E-14  0.100000E-11  0.406592E-10  0.876000E-
   5    8760.0   0.339019E-05  0.219585E-03  0.000000E+00  0.000000E+00  0.341048E-05  0.369218E-01  0.594731E-
   6    8760.0   0.999999E-10  0.438050E-14  0.876000E-06  0.191373E-10  0.100000E-09  0.500069E-08  0.875980E-
   7    8760.0   0.100000E-11  0.356174E-13  0.876000E-08  0.156580E-14  0.100000E-11  0.406592E-10  0.876000E-
   8    8760.0   0.999999E-10  0.438050E-14  0.876000E-06  0.191373E-10  0.100000E-09  0.500069E-08  0.875980E-
   9    8760.0   0.100000E-11  0.356174E-13  0.876000E-08  0.156580E-14  0.100000E-11  0.406592E-10  0.876000E-
  10    8760.0   0.640849E-09  0.905262E-05  0.000000E+00  0.000000E+00  0.640850E-09  0.506290E+00  0.178303E-
  11    8760.0   0.801790E-06  0.383265E-05  0.000000E+00  0.000000E+00  0.804365E-06  0.119740E-02  0.320081E-
  12    8760.0   0.999999E-10  0.438050E-14  0.876000E-06  0.191373E-10  0.100000E-09  0.500069E-08  0.875980E-
  13    8760.0   0.100000E-11  0.356174E-13  0.876000E-08  0.156580E-14  0.100000E-11  0.406592E-10  0.876000E-
  14    8760.0   0.999999E-10  0.438050E-14  0.876000E-06  0.191373E-10  0.100000E-09  0.500069E-08  0.875980E-
  15    8760.0   0.100000E-11  0.356174E-13  0.876000E-08  0.156580E-14  0.100000E-11  0.406592E-10  0.876000E-
  16    8760.0   0.735404E-05  0.145963E-05  0.000000E+00  0.000000E+00  0.780068E-05  0.254929E-04  0.672565E-
  17    8760.0   0.282559E-04  0.282559E-03  0.000000E+00  0.000000E+00  0.294128E-04  0.718401E-02  0.393317E-
  18    8760.0   0.999999E-10  0.438050E-14  0.876000E-06  0.191373E-10  0.100000E-09  0.500069E-08  0.875980E-
  19    8760.0   0.100000E-11  0.356174E-13  0.876000E-08  0.156580E-14  0.100000E-11  0.406592E-10  0.876000E-
  20    8760.0   0.249245E-05  0.155778E-04  0.000000E+00  0.000000E+00  0.251438E-05  0.178541E-02  0.872507E-
  21    8760.0   0.249245E-05  0.155778E-04  0.000000E+00  0.000000E+00  0.251438E-05  0.178541E-02  0.872507E-
  22    8760.0   0.298233E-05  0.294572E-05  0.261584E-01  0.202569E-01  0.300000E-05  0.500000E-03  0.589145E-
  23    8760.0   0.199996E-08  0.210128E-11  0.175198E-04  0.920530E-08  0.200000E-08  0.120000E-06  0.175106E-
  24    8760.0   0.200000E-09  0.175206E-13  0.175200E-05  0.767417E-10  0.200000E-09  0.100000E-07  0.175192E-
  25    8760.0   0.247796E-05  0.220423E-05  0.217705E-01  0.129536E-01  0.250000E-05  0.250000E-03  0.881591E-

UT       t          w*(t)         Lambda*(t)          Q*(t)
```

```
 8760.0   0.214083E-06   0.214156E-06   0.337324E-03
 8760.0   0.206719E-06   0.206823E-06   0.504825E-03
 8760.0   0.283768E-06   0.283834E-06   0.231721E-03
 8760.0   0.346592E-06   0.346712E-06   0.345784E-03

  t          LAMBDA(S)          W(s)             Q(S)MIN          Q(S)MAX          Q(S)MEAN
50.0   0.591583E-11   0.591583E-11   0.255554E-08   0.257222E-08   0.255943E-08

OF EXECUTION
TIME: 39.83 ELAPSED TIME: 2:36.74

B/BATCH

AJL Another job is still logged-in under [15100,150064]]
23  User MANOJ KUMAR [15100,150064]
ed-off TTY115  at 13:49:07  on 29-Nov-85
ime: 0:00:40, KCS:887, Connect time: 0:02:45
 Reads:217, Writes:3
```

```
ON version 102(2067) running CT sequence 6011 in stream 1
t from DSKC:CT.CTL[15100,150064]
ut to  DSKC:CT.LOG[15100,150064]
parameters
:00:01:00   Core:100P   Unique:YES   Restart:YES   Output:NOLOG

GIN 15100/150064 /DEFER/SPOOL:ALL/TIME:60/CORE:100P/LOCATE:10/NAME:"MANOJ KUMAR"
 31 I I T KANPUR 603A(3) TTY116
JSP Other jobs same PPN:18]
3     13-Dec-86        Sat
 have some mail. Type *.BOX.

 CT.FOR
:38:23]
      C         PROGRAM NAME MINCELS
      C
      C         PROGRAM FOR MINIMAL CUT AND PATH SETS FOR A ELECTRICAL SYSTEM

      C
                INTEGER CO
                COMMON CO(7,50),INN(50),ICOUNT(50)
                COMMON/PATH/MPT(25,40),MPTJ(25,40),MPTD(25,40),MP(100,15),NP,LP
                DIMENSION INODE(10)
                READ(22,100)NODE
      100       FORMAT(I3)
                DO 20 IR=1,7
                IS=1
      15        READ(22,200)(CO(IR,IC),IC=IS,IS+19)
      200       FORMAT(20I3)
                IF(CO(IR,IS+19).EQ.0)GOTO 20
                IS=IS+20
                GOTO 15
      20        CONTINUE
                DO 25 I=1,7
                DO 25 L=1,50
      25        IF(CO(I,L).EQ.0)CO(I,L)=-1
                READ(22,100)NOS
                READ(22,200)(INODE(L),L=1,NOS)
                WRITE(5,300)(INODE(L),L=1,NOS)
      300       FORMAT(/,1X,'SOURCE NODES: ',20I3)
                DO 30 I=1,NOS
      30        INN(INODE(I))=1
                CALL TREE(NODE)
                CALL BCUT
                STOP
                END

      C         &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
                SUBROUTINE TREE(NODE)
                INTEGER CO
                COMMON CO(7,50),INN(50),ICOUNT(50)
                COMMON/PATH/MPT(25,40),MPTJ(25,40),MPTD(25,40),MP(100,15),NP,LP
      C         ----------OBTAINING BASIC PATH TREES------------
                LEV=0
                MPT(1,1)=NODE
      20        LEV=LEV+1
                IF(MPT(LEV,1).EQ.0)GOTO 70
                LL=0
                LN=0
      30        LL=LL+1
                IF(MPT(LEV,LL).EQ.0)GOTO 20
      IF(INN(MPT(LEV,LL)).EQ.1)GOTO 30
      LNC=0
 32   LN=LN+1
      MPP=MPT(LEV,LL)
 35   LNC=LNC+1
      IF(INN(CO(LNC,MPP)).EQ.1)GOTO 55
      IF(CO(LNC,MPP).NE.-1)GOTO 40
      LN=LN-1
      MPTD(LEV,LL)=LN
      GOTO 30
 40   IF(LEV.EQ.1)GOTO 55
      ITT=1
      LLT=MPTU(LEV,LL)
      MPPT=MPT(LEV-1,LLT)
 45   IF(CO(LNC,MPP).EQ.MPPT)GOTO 35
      IF(LLT.NE.1)GOTO 46
 43   DO 44 IT=1,MPTD(LEV-ITT,LLT)
 44   IF(CO(LNC,MPP).EQ.MPT(LEV-ITT+1,IT))GOTO 35
      GOTO 52
 46   IM=1
 47   IF(MPTD(LEV-ITT,LLT-IM).GT.0)GOTO 48
      IM=IM+1
      IF(LLT.EQ.IM)GOTO 43
      GOTO 47
 48   DO 50 IT=MPTD(LEV-ITT,LLT-IM)+1,MPTD(LEV-ITT,LLT)
 50   IF(CO(LNC,MPP).EQ.MPT(LEV-ITT+1,IT))GOTO 35
 52   ITT=ITT+1
      IF(ITT.EQ.LEV)GOTO 55
      LLT=MPTU(LEV-ITT+1,LLT)
      MPPT=MPT(LEV-ITT,LLT)
      GOTO 45
 55   MPT(LEV+1,LN)=CO(LNC,MPP)
      MPTU(LEV+1,LN)=LL
      GOTO 32
C     ------------------OBTAINING BASIC MINIMAL PATH FROM TREE------------
 70   WRITE(5,1000)NODE
 1000 FORMAT(/,' FOR SINK NODE:',I3,//,' THE BASIC MINIMAL PATHS :')
      NP=0
      DO 90 I=LEV-1,2,-1
      N=1
 80   IF(MPT(I,N).EQ.1)GOTO 90
      IF(INN(MPT(I,N)).EQ.1)GOTO 100
 85   N=N+1
      GOTO 80
 90   CONTINUE
      RETURN
 100  NP=NP+1
      LP=1
      LL=I
      LN=N
 110  MP(NP,LP)=MPT(LL,LN)
      ICOUNT(MP(NP,LP))=ICOUNT(MP(NP,LP))+1
      IF(LL.GT.1)GOTO 115
      WRITE(5,1010)NP,(MP(NP,NN),NN=1,LP)
 1010 FORMAT(11X,' BASIC PATH ',I4,' :-',20I4)
      GOTO 85
 115  LN=MPTU(LL,LN)
      LL=LL-1
      LP=LP+1
      GOTO 110
      END
C     &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
C
      SUBROUTINE BCUT
      INTEGER CO
      COMMON/PATH/MPT(25,40),MPTU(25,40),MPTD(25,40),MP(100,15),NP,LP
      COMMON CO(7,50),INN(50),ICOUNT(50)
      DIMENSION LFIRST(20),LSEC(40,2),LTHIRD(50,3),LFOUR(80,4)
      DIMENSION LT(15),INP(30,5),INC(50),MT(25),LM(45),D2I(15,20)
      DIMENSION D3I(25,40)
C     ------------FIRST ORDER CUTSETS-------------

      N1=0
      NT=0
      DO 20 I=1,LP
      IF(ICOUNT(MP(NP,I)).EQ.NP)GOTO 10
      NT=NT+1
      INC(MP(NP,I))=-NT
      LT(NT)=MP(NP,I)
      GOTO 20
10    INC(MP(NP,I))=-20
      N1=N1+1
      LFIRST(N1)=MP(NP,I)
20    CONTINUE
      WRITE(5,200)(LFIRST(K),K=1,N1)
200   FORMAT(1X,' FIRST ORDER CUTSET : ',20I3)
      DO 50 I=1,NP
      NI=0
      NO=0
40    NO=NO+1
      IF(MP(I,NO).EQ.0)GOTO 48
      IF(INC(MP(I,NO)).LT.0)GOTO 44
      NI=NI+1
      MT(NI)=MP(I,NO)
      INC(MP(I,NO))=1
      MP(I,NO)=0
      GOTO 40
44    IF(INC(MP(I,NO)).EQ.-20)GOTO 46
      INP(I,INC(MP(I,NO))*-1)=1
46    MP(I,NO)=0
      GOTO 40
48    DO 50 K=1,NI
      MP(I,K)=MT(K)
50    CONTINUE
      NM=0
      DO 55 J=1,NT
      NM=NM+1
55    LM(NM)=LT(J)
      DO 60 J=1,50
      IF(INC(J).NE.1)GOTO 60
      NM=NM+1
      LM(NM)=J
60    CONTINUE
      IF(NP.LT.2)GOTO 130
C     ---------------SECOND ORDER CUTSET-----------------
      N2=0
      DO 80 I=1,NT
      NL2=0
      DO 80 J=I+1,NM
      IF(LT(I).EQ.LM(J))GOTO 80
      IF(ICOUNT(LT(I))+ICOUNT(LM(J)).LT.NP)GOTO 80
      DO 75 K=1,NP
      M=0
      IF(INP(K,I).EQ.1)GOTO 75
      IF(INC(LM(J)).GT.0)GOTO 70
      IF(INP(K,INC(LM(J))*-1).EQ.1)GOTO 75
70    M=M+1
      IF(MP(K,M).EQ.0)GOTO 80
      IF(MP(K,M).EQ.LM(J))GOTO 75
      GOTO 70
75    CONTINUE
      N2=N2+1
      LSEC(N2,1)=LT(I)
      LSEC(N2,2)=LM(J)
      NL2=NL2+1
      L2I(I,NL2)=LM(J)
80    CONTINUE
      IF(N2.EQ.0)GOTO 95
      WRITE(5,201)
201   FORMAT(1X,' SECOND ORDER CUTSET : ')
      DO 90 LL=1,N2
90    WRITE(5,202)LL,LSEC(LL,1),LSEC(LL,2)
202   FORMAT(14X,I3,':',2I4)
95    IF(NP.LT.3)GOTO 130
C     ---------------THIRD ORDER CUTSET---------------
      N3=0
      DO 110 I=1,NT
      NL3=0
      DO 110 J=I+1,NM
      DO 110 K=J+1,NM
      IF(ICOUNT(LT(I))+ICOUNT(LM(J))+ICOUNT(LM(K)).LT.NP)GOTO 110
      N=0
100   N=N+1
      IF(L2I(I,N).EQ.0)GOTO 101
      IF(L2I(I,N).EQ.LM(J).OR.L2I(I,N).EQ.LM(K))GOTO 110
      GOTO 100
101   IF(J.GT.NT)GOTO 102
      NMT=0
99    NMT=NMT+1
      IF(L2I(J,NMT).EQ.0)GOTO 102
      IF(L2I(J,NMT).EQ.LM(K))GOTO 110
      GOTO 99
102   DO 105 L=1,NP
      IF(INP(L,I).EQ.1)GOTO 105
      M=0
      IF(J.GT.NT)GOTO 103
      IF(INP(L,INC(LM(J))*-1).EQ.1)GOTO 105
103   IF(K.GT.NT)GOTO 104
      IF(INP(L,INC(LM(K))*-1).EQ.1)GOTO 105
104   M=M+1
      IF(MP(L,M).EQ.0)GOTO 110
      IF(MP(L,M).EQ.LM(J).OR.MP(L,M).EQ.LM(K))GOTO 105
      GOTO 104
105   CONTINUE
      N3=N3+1
      LTHIRD(N3,1)=LT(I)
      LTHIRD(N3,2)=LM(J)
      LTHIRD(N3,3)=LM(K)
      NL3=NL3+1
      L3I(I,NL3)=LM(J)*100+LM(K)
110   CONTINUE
      IF(N3.EQ.0)GOTO 130
      WRITE(5,203)
203   FORMAT(1X,' THIRD ORDER CUTSET : ')
      DO 120 ML=1,N3
120   WRITE(5,204)ML,(LTHIRD(ML,KL),KL=1,3)
204   FORMAT(14X,I3,':',3I4)
130   RETURN
```

```
23700            END

.TY FOR22.DAT
[14:38:31]
  1
  2     1     1     2     2     5     3     5     4     4     7     9
  3     4     5     5     4     7     6     7    10     8     8    11
        5     7     9     6     0     8    10    12     9    10
                   10     8     0    11    11     0    11    12



  1
 12

.EX CT.FOR
[14:38:31]
LINK:    Loading
[LNKXCT CT execution]

SOURCE NODES:  12

FOR SINK NODE:  1

THE BASIC MINIMAL PATHS :
                BASIC PATH    1 :-   12    9   10   11    7    6    5    2    1
                BASIC PATH    2 :-   12   11    7    5    5    2    1
                BASIC PATH    3 :-   12    9   10    8    5    2    1
                BASIC PATH    4 :-   12   11   10    4    5    3    1
                BASIC PATH    5 :-   12    9   10    8    5    3    1
                BASIC PATH    6 :-   12    9   10    8    7    3    1
                BASIC PATH    7 :-   12    9   10   11    7    3    1
                BASIC PATH    8 :-   12   11   10    4    2    1
                BASIC PATH    9 :-   12   11    8    5    2    1
                BASIC PATH   10 :-   12    9    4    5    3    1
                BASIC PATH   11 :-   12   11    8    5    3    1
                BASIC PATH   12 :-   12   11    8    5    2    1
                BASIC PATH   13 :-   12    9    4    5    3    1
                BASIC PATH   14 :-   12    9    4    2    1
                BASIC PATH   15 :-   12   11    7    3    1
 FIRST ORDER CUTSET :  12   1
 SECOND ORDER CUTSET :
                   1:  11    9
                   2:   3    2
 THIRD ORDER CUTSET :
                   1:  11    4    8
                   2:  11    4   10
                   3:   7    2    5
                   4:   7    4    5
                   5:   7    4    8
                   6:   3    4    5
STOP

END OF EXECUTION
CPU TIME: 0.20  ELAPSED TIME: 1.18
EXIT

.KJOB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 31  User MANOJ KUMAR [15100,150064]
Logged-off TTY116  at 14:38:34  on 13-Dec-85
Runtime: 0:00:01, KCS:16, Connect time: 0:00:09
```

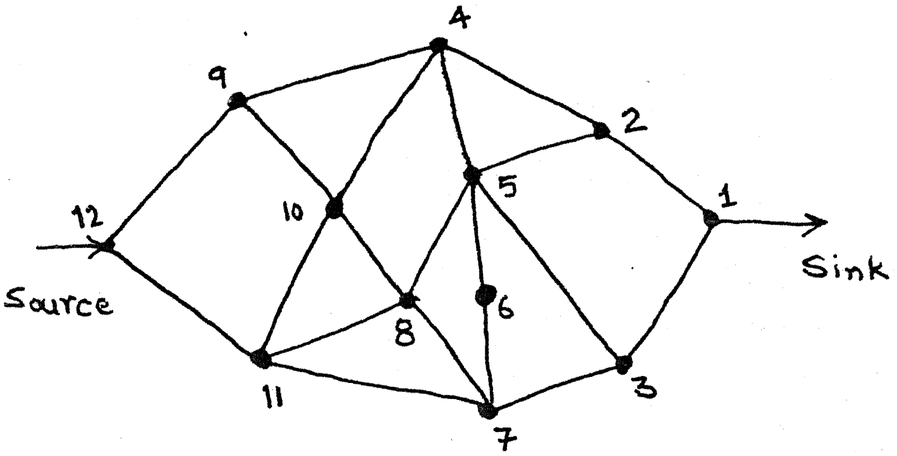


Figure : Example ~~for~~ MINCELS

```
TCON version 102(2067) running LAST sequence 2445 in stream 1
put from DSKC:LAST.CTL[15100,150064]
tput to  DSKC:LAST.LOG[15100,150064]
b parameters
me:00:01:00    Core:100P    Unique:YES    Restart:YES    Output:NOLOG


LOGIN 15100/150064 /DEFER/SPOOL:ALL/TIME:60/CORE:100P/LOCATE:10/NAME:"MANOJ KUMAR"
OB 17 I I T KANPUR 603A(3) TTY115
LGNJSP Other jobs same PPN:24]
323      29-Nov-86         Sat

TY LAST.FOR
3:23:24]
C              ################################################
C              PROGRAM NAME:"MODTREE"
C              ################################################
C              THIS PROGRAM MODIFIES THE FAULT TREE TO TAKE CARE OF
C              XOR and NOT GATES.XOR GATE IS SHOWN BY ASSIGNING TO
C              THE GATE NO. LARGER THAN 100.NOT GATE IS INPUTTED BY
C              ASSIGNING IT A NEGATIVE NO.THUS,NOR and NAND GATES
C              ARE ALSO TAKEN CARE OF.OUTPUT CONSISTS OF TREE HAV-
C              ING ONLY OR ,AND GATES and BASIC EVENTS EITHER NEGATED
C              OR UN-NEGATED.
C              M OUT OF N KIND OF GATES ARE TAKEN CARE OF AT QUANTIFI-
C              CATION STAGE.
C              THE PROGRAM UTILISES DE MORGAN'S LAW AND GATE EQUIVALENCE.
C
C              A@B=Ā.B+A.B̄    A̅.̅B̅=Ā+B̄   A̅+̅B̅=Ā.B̄
C              ***********************************************
               INTEGER OK,GN(100,100),S,B,AO,OR
C              L=LEVEL OF TREE. OR=NO. OF OR GATES IN THE TREE
C              MO=MAX. ORDER OF CUTSETS WANTED.
               READ(22,1)L,OR,MO
1              FORMAT(1X,3(I3,1X))
               B=0
C              INPUT FOR FAULT TREE
               DO 10 I=1,100
               READ(22,2),(GN(I,J),J=1,L+1)
2              FORMAT(100I4)
               IF(GN(I,1).EQ.0)GOTO 20
               B=B+1
10             CONTINUE
C              B=NO. OF BASIC EVENTS.
20             S=0
               JP=0
               LS=L
               DO 130 J=1,100
               DO 140 I=1,100
               IF(I.GT.B)GOTO 130
               IF(J.GT.L)GOTO 135
               IF(GN(I,J).GT.100)GOTO 145
               DO 25 J1=J,L
               IF(GN(I,J1).EQ.0)GOTO 28
               AO=AO+1
25             CONTINUE
C              AO=LEVEL OF SUB-BRANCH CONTAINING THE BASIC EVENT.
28             IF(AO.NE.L)GOTO 29
               LS=L
               L=L+1
29             IF(GN(I,J).EQ.JP)GOTO 32
               S=0
```

```
      JP=GN(I,J)
32    IF(S.NE.0)GOTO 35
      CK=GN(I,J+1)
35    IP1=I+1
      IP2=I+2
      JP1=J+1
      JP2=J+2
      LP1=L+1
      GN(I,J)=1
      IF(CK.NE.GN(I,JP1))GOTO 40
      IF(GN(I,LS).EQ.0)LPP=L
      IF(GN(I,LS).NE.0)LPP=L+1
      DO 80 J2=J,LPP
      M=LPP-J2+J
      IF(M.LT.JP2)GOTO 82
      GN(I,M)=GN(I,M-1)
80    CONTINUE
82    GN(I,J+1)=2
      GN(I,JP2)=-1*GN(I,JP2)
      B=B+1
      DO 170 I2=IP2,B
      DO 160 J2=1,LP1
      M=B-I2+IP2
      GN(M,J2)=GN(M-1,J2)
160   CONTINUE
170   CONTINUE
      DO 175 J2=1,LP1
      GN(IP1,J2)=GN(I,J2)
175   CONTINUE
      GN(IP1,JP1)=4
      GN(IP1,JP2)=-1*GN(IP1,JP2)
      S=1
      GOTO 145
40    IF(GN(I,LS).EQ.0)LPP=L
      IF(GN(I,LS).NE.0)LPP=L+1
      DO 45 J2=J,LPP
      M=LPP-J2+J
      IF(M.LT.JP2)GOTO 48
      GN(I,M)=GN(I,M-1)
45    CONTINUE
48    GN(I,J+1)=2
      B=B+1
      DO 70 I2=IP2,B
      DO 60 J2=1,LP1
      M=B-I2+IP2
      GN(M,J2)=GN(M-1,J2)
60    CONTINUE
70    CONTINUE
      DO 75 J2=1,LP1
      GN(IP1,J2)=GN(I,J2)
75    CONTINUE
      GN(IP1,JP1)=4
      GN(IP1,JP2)=-1*GN(IP1,JP2)
      S=1
145   CONTINUE
140   CONTINUE
130   CONTINUE
C     RESTRUCTURING OF TREE TO MAX ORDER AND LEVEL OF MODIFIED
C     TREE.
135   DO 300 I=1,B
      DO 310 J=1,L
      IF(GN(I,L+1).NE.0)GOTO 300
      J1=L-J+1
```

```
              IF(GN(I,J1).EQ.0)GOTO 310
              GN(I,L+1)=GN(I,J1)
              GN(I,J1)=0
      310     CONTINUE
      300     CONTINUE
      C       FOLLOWING TAKES CARE OF NOR,NOT,NAND GATES.
              DO 200 I=1,B
              DO 210 J=1,L
              IF(GN(I,J).GE.0)GOTO 210
              IF(MOD(GN(I,J),2).NE.0)GOTO 230
              GN(I,J)=1
              GN(I,J+1)=-1*GN(I,J+1)
              GOTO 210
      230     GN(I,J)=2
              GN(I,J+1)=-1*GN(I,J+1)
      210     CONTINUE
      200     CONTINUE
      C       OUTPUTTING THE MODIFIED FAULT TREE.
              WRITE(5,240)L,OR,MO
      240     FORMAT(1X,3(I3,1X))
              DO 260 I=1,B
              WRITE(5,250),(GN(I,J),J=1,L+1)
      250     FORMAT(1X,100I4)
      260     CONTINUE
              STOP
              END

.TY FOR22.DAT
[3:23:27]
    1    1   10
  210    1
  210    2
    0

.EX LAST.FOR
[3:23:28]
LINK:   Loading
[LNKXCT LAST execution]

   2   1  10
   1   2  -1
   1   4   1
   1   2   2
   1   4  -2
STOP

END OF EXECUTION
CPU TIME: 0.04  ELAPSED TIME: 0.44
EXIT

.KJOB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 17  User MANOJ KUMAR [15100,150064]
Logged-off TTY115  at  3:23:31  on 29-Nov-85
Runtime: 0:00:00, KCS:10, Connect time: 0:00:11
Disk Reads:234, Writes:3
BATCON version 102(2067) running LAST sequence 2447 in stream 1
Input from DSKC:LAST.CTL[15100,150064]
Output to  DSKC:LAST.LOG[15100,150064]
Job parameters
Time:00:01:00   Core:100P   Unique:YES   Restart:YES   Output:NOLOG
```

```
.LOGIN 15100/150064 /DEFER/SPOOL:ALL/TIME:60/CORE:100P/LOCATE:10/NAME:"MANOJ KUMAR
JOB 13 I I T KANPUR 603A(3) TTY115
[LGNJSP Other jobs same PPN:24]
0325     29-Nov-86        Sat

.TY FOR22.DAT
[3:25:13]
   3    3    5
   2    4    0    1
   2    4    5    2
   2    4    5    3
   2  201    0    4
   2  201    1    5
   2  201    1    6
   0

.EX LAST.FOR
[3:25:13]
LINK:   Loading
[LNKXCT LAST execution]

   4    3    5
    2    4    0    0    1
    2    4    5    0    2
    2    4    5    0    3
    2    1    2    0   -4
    2    1    4    0    4
    2    1    2    1    5
    2    1    4    2   -5
    2    1    2    1    6
    2    1    4    2   -6
STOP

END OF EXECUTION
CPU TIME: 0.06  ELAPSED TIME: 0.36
EXIT

.KJOB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 13  User MANOJ KUMAR [15100,150064]
Logged-off TTY115  at  3:25:15  on 29-Nov-86
Runtime: 0:00:00, KCS:7, Connect time: 0:00:07
Disk Reads:181, Writes:3
```

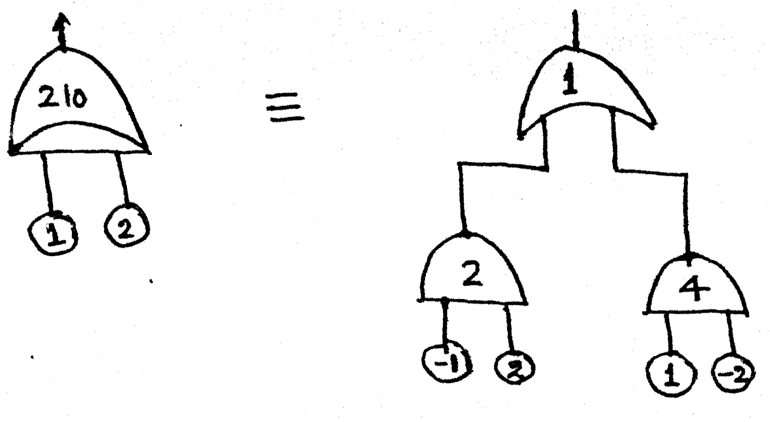


Figure : Example 1

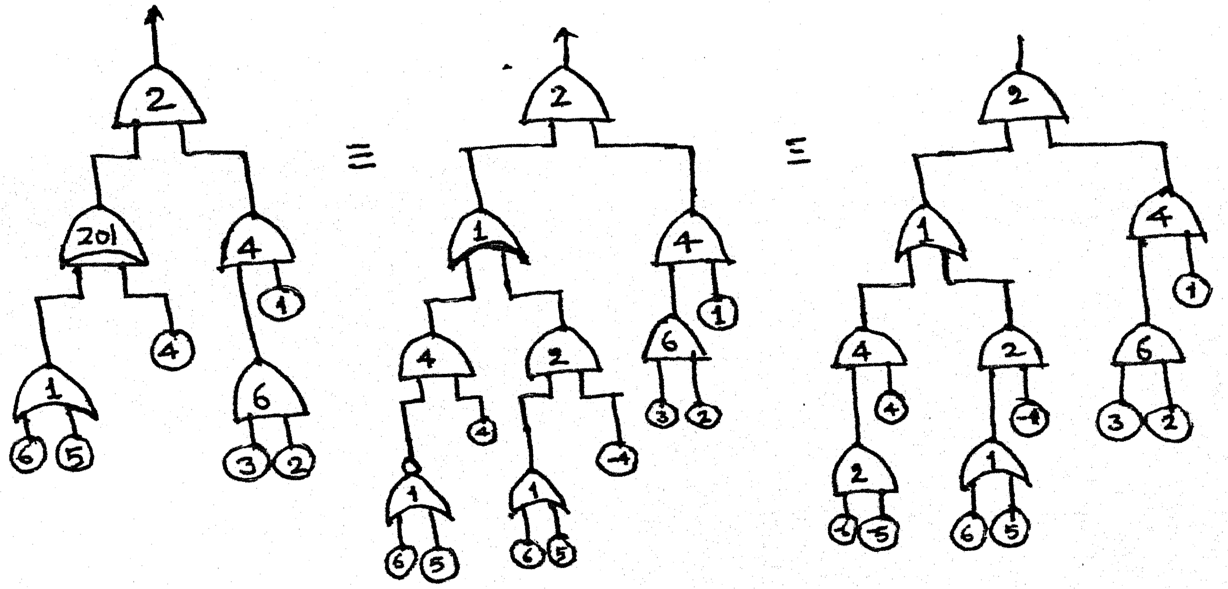


Figure : Example 2

For MODTREE

NETP-1987-M-KUM-AUT